



Universidad de las Ciencias Informáticas

Facultad 4

*Herramienta de Administración del producto “Mis Mejores Cuentos”.*

*Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.*

**Autores**

*Yoanny Quintero Sánchez.*

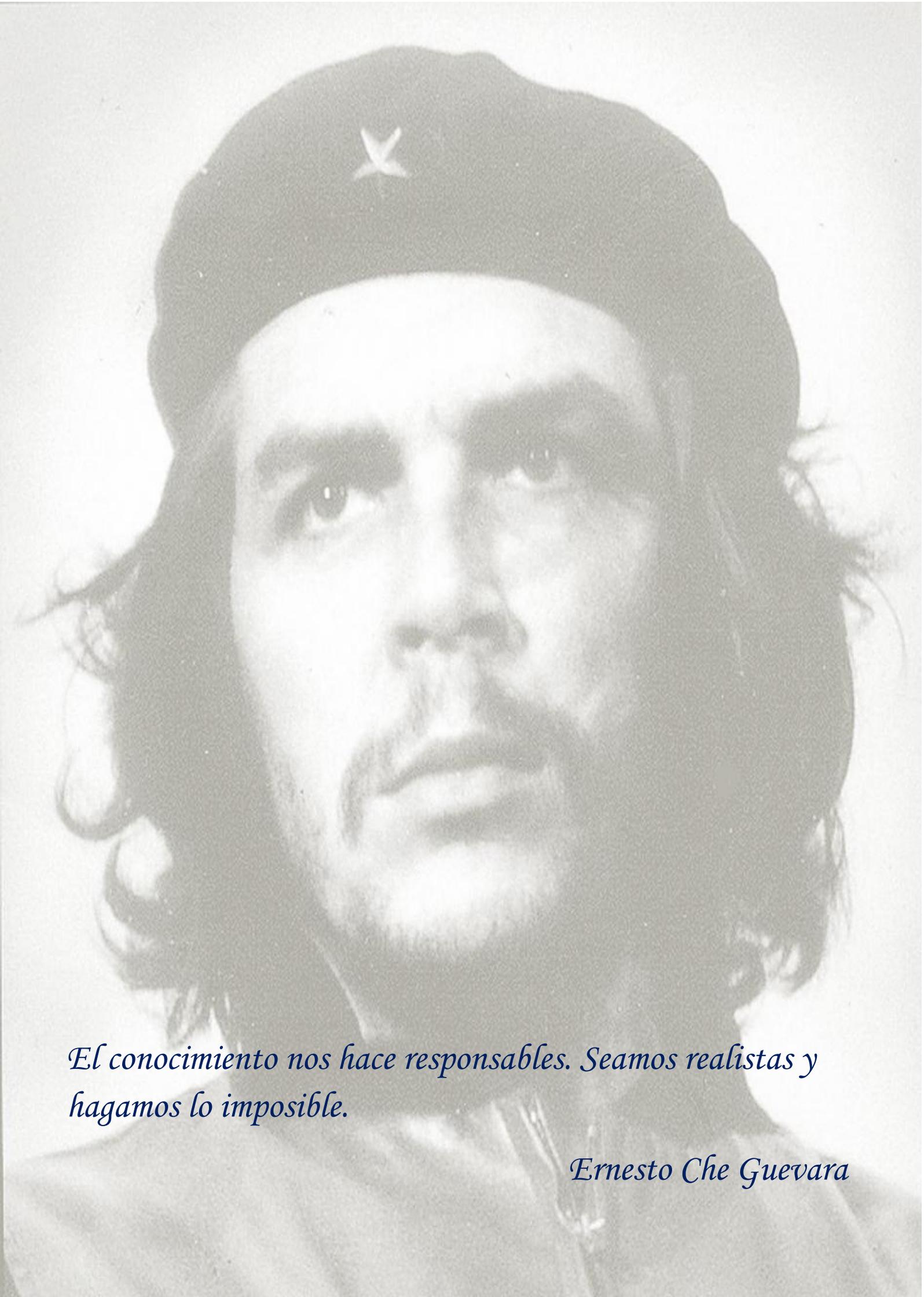
*Gladys Adrianet Trinchet Castro.*

**Tutora**

*Ing. Lisandra Guibert Estrada.*

La Habana

“Año 53 de la Revolución”



*El conocimiento nos hace responsables. Seamos realistas y hagamos lo imposible.*

*Ernesto Che Guevara*

## Declaración de Autoría

Declaramos ser los autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Yoanny Quintero Sánchez  
(Autor)

\_\_\_\_\_  
Gladys Adrianet Trinchet Castro  
(Autora)

\_\_\_\_\_  
Ing. Lisandra Guibert Estrada  
(Tutora)

# Agradecimientos

## **Comunes:**

*A Fidel y a la Revolución por permitir que este sueño se convierta en realidad.*

*A la Universidad de las Ciencias Informáticas y especialmente a la facultad de los octavitos por formarnos como futuros profesionales.*

*A nuestra tutora Lisandra por estar siempre presente cuando la necesitamos y soportar a sus estresantes tesis en los cortes.*

*A los profes Isyed y Joe por ser nuestros tutores cuando Lisandra estuvo en Venezuela y lograr que unieran la tesis.*

*A Yasmani (El Súper) por todo su apoyo.*

## **De La Nena:**

*A mi mamita por siempre estar a mi lado, complacerme en todo e intentar hacer de mí una persona mejor cada día.*

*A mi abuela Aida y mi tía Eneidis, mis otras madres, por dedicarme tanto tiempo de sus vidas.*

*A mi padre por apoyarme siempre y consentirme en todo.*

*A Yuslei por ser el hermano que siempre quise tener y mi ejemplo a seguir.*

*A mi hermana que aunque lejos ha estado al tanto de mí.*

*A mi abuelo, mis tíos, mis primos y a toda mi familia por su apoyo en cada etapa de mi vida, que sin ellos habría sido imposible llegar hasta aquí.*

*A mis nenitas que tanto quiero Yamiselis y Vanessita las más pequeñas de la familia que sean mejores de lo que he sido yo.*

*A mi súper compañero de tesis (El Yoa), el único que he tenido pero no creo que exista uno mejor.*

*A Miguel Ernesto por confiar todo este tiempo en mí y soportar mis malacrianzas.*

*A Jose (Pepe) por lo mucho que me ha ayudado en los 5 años de mi carrera.*

*A Rigo por ser la persona que me enseñó a dar los primeros pasos cuando llegué a la UCI.*

*A la profe Thaimí por estar pendiente siempre de todo lo que sucede con nosotros.*

*A mi primo Eric por su disposición y ayuda en la revisión del documento de tesis.*

*A mis amigos de Jiguaní Lisdanis, Rotsana L, Sergio, Livan, Angel (Papo), Yadira, Alain y Alejandro.*

*A todos los amigos de estos cinco años, algunos que se fueron y a los que aún quedan acá, Kirenia, Arleen, Queen Ely, Darián, Nayla, Ramón, El Migue, Yaulio, Samuel y El Michia que de cada uno he aprendido algo.*

*A las pamples que es imposible dejar de mencionar la Yessy, Lili y Sasy por soportarme cada momento del día a día.*

*GRACIAS A TODOS POR EL SIMPLE HECHO DE EXISTIR PARA MI.*

## **De Yoanny:**

*A mis padres (Humberto y Cristina) por su sacrificio y su apoyo incondicional en los 5 años de la carrera, por escucharme siempre y estar a mi lado en los momentos en que más los he necesitado. Gracias por estar tan orgullosos de mí, ustedes saben que nunca los defraudaré. Los quiero mucho.*

*A mi hermanita (Pilar) que yo la quiero mucho y siempre me he esforzado para ser un ejemplo para ella, gracias por tu cariño y confianza.*

*A mi novia (Rosali) por estar siempre a mi lado en estos dos últimos años de mi carrera y ayudarme en todo lo que he necesitado, por darme sus consejos y sus opiniones en este trabajo de diploma. Te quiero mucho mi chiquitica linda.*

*A mi tía (Cary) por todo el apoyo que me ha brindado durante toda la carrera, que si no fuera por ella hubiera roto el record de más guaguas montadas en 5 años. Gracias por estar ahí cada vez que te necesito y por sentirte orgullosa de tus sobrinos y por querernos mucho.*

*A mi tía (Ismary) que nunca me ha dado la espalda cuando más la he necesitado y que siempre ha estado pendiente de mí y de todo lo que me pueda hacer falta.*

*A mis abuelos que siempre, aunque ellos no lo crean me enseñan algo nuevo cada día.*

*A dos personas que también quiero mucho (Julio y Maritza) que me han brindado su apoyo y su cariño.*

*A mi compañera de tesis (La Nena) que siempre estaba ahí para corregirme todos los errores y estar pendiente siempre del más mínimo detalle. Si me encuentro una mejor compañera de tesis que Adrianet la voy a tumbar que seguro es de cartón.*

*A mi amigo y hermano (Osvaldo Aguilar) que me ayudó mucho a sentar las bases de lo que hoy es el Módulo de Administración de Cuentos, para el producto “mis mejores cuentos”. Gracias hermano por el apoyo que me diste.*

*A todos mis amigos que llevan toda la vida arrastrando conmigo y que considero mis hermanos (Julito, Yanquiel, Ismael, Dino, Albertón, Roberto, Karel, Landy, Chino). Que me han brindado su apoyo siempre que los he necesitado.*

*A todos mis compañeros de la carrera y especialmente los de mi apto (Napo, Metra, Jorgito, Ariel) por sus críticas constructivas que me ayudaron mucho en la realización de este trabajo de diploma.*

*A mis suegros (Evidio y Julia) que de una forma u otra también me han ayudado mucho.*

*A mi Santa Bárbara que siempre me protege y me guía por el buen camino.*

*A todas las personas que aportaron su granito en la realización de este trabajo y que no las menciono aquí, les estoy eternamente agradecido.*

**Yo@.**

*A nuestros padres, familia y amigos por ser los mejores del mundo y brindarnos su apoyo y amor incondicional.*

## Resumen

En la actualidad, debido a la gran cantidad de información que se procesa diariamente, las herramientas de administración o sistemas de gestión de contenidos para aplicaciones Web son muy importantes, ya que con su uso se agiliza el trabajo de edición y procesamiento de información. El producto “Mis Mejores Cuentos” desarrollado por el proyecto “Producción de Recursos Didácticos” de la Universidad de las Ciencias Informáticas (UCI) necesitaba una herramienta de gestión de contenido para agilizar el manejo y publicación de su información, por tanto se lleva a cabo la implementación de la Herramienta de Administración para la multimedia Web “Mis Mejores Cuentos”.

En el presente trabajo se hace un análisis de las herramientas de administración de contenidos Web y de sus últimas tendencias. Se realiza un estudio y selección de las tecnologías adecuadas para el desarrollo de la herramienta, se levantan los requisitos y se identifican las diferentes funcionalidades que se deben cumplir. Se realiza el diseño y partiendo de este se procede a la implementación y validación de la herramienta obteniéndose excelentes resultados según las pruebas realizadas al software.

## **Abstract**

Today, due to the large amount of information that is processed daily, the management tools or content management systems for Web applications are very important, and that their use is quick work of editing and processing of information. The product "Mis Mejores Cuentos" developed by the project "Producción de Recursos Didácticos" at the University of Informatics Sciences (UCI) needed a content management tool to streamline the management and publishing of information, therefore carried out the implementation of the Administration Tool for Web multimedia "Mis Mejores Cuentos".

In this work we study tools for Web content management and its latest trends. We performed a study and selection of appropriate technologies for development tool; a requirement is lifted, and identifies the different features that must be met. Design is done and from this we proceed to the implementation and validation of the tool with excellent results as tested software.

Declaración de Autoría .....	2
Agradecimientos.....	3
Dedicatoria .....	5
Resumen.....	6
Abstract.....	7
Índice .....	8
Introducción.....	11
Capítulo 1 .....	15
1. Introducción. ....	15
1.2 Estado del arte. ....	15
1.2.1 Software Educativo. ....	15
1.2.2 Sistemas de Gestión de Contenidos (CMS). ....	15
1.2.3 ¿Qué es la Gestión de Contenidos Educativos? .....	17
1.2.4 ¿Qué es un LMS? .....	17
1.2.5 ¿Qué es un LCMS?.....	17
1.2.6 Herramientas de Administración Web. ....	17
1.2.7 Herramientas Cubanas. ....	19
1.3 Metodologías de desarrollo. ....	20
1.3.1 Programación Extrema (XP).....	21
1.3.2 SCRUM.....	22
1.3.3 Microsoft Solutions Framework (MSF).....	22
1.3.4 Metodología de Administración de Relaciones (RMM) .....	23
1.3.5 RUP (Rational Unified Process) .....	24
1.4 Lenguajes de modelado. ....	25
1.4.1 Lenguaje Unificado de modelado (UML). ....	25
1.4.2 Lenguaje para modelar Aplicaciones Multimedia (OMMMA-L) .....	26
1.4.3 Lenguaje para Modelar Aplicaciones Educativas (ApEM-L). ....	26
1.5 Herramientas de ingeniería asistida por computadoras (Computer Aided Software Engineering (CASE)) .....	27
1.5.1 Visual Paradigm (VP) .....	27
1.5.2 Rational Rose Enterprise.....	28
1.5.3 Umbrello UML Modeller.....	28
1.6 Lenguajes de desarrollo web.....	29
1.6.1 Lenguaje HTML.....	29
1.6.2 Lenguaje Javascript .....	29

1.6.3	Lenguaje Python .....	30
1.6.4	Lenguaje PHP .....	30
1.6.5	Lenguaje ASP .....	31
1.7	Framework de desarrollo.....	32
1.7.1	Symfony.....	32
1.7.2	ExtJs.....	33
1.7.3	JQuery: .....	33
1.7.4	Prototype.....	33
1.7.5	Scriptaculous.....	33
1.8	Entorno de Desarrollo Integrado (IDE) .....	33
1.8.1	NetBeans .....	34
1.8.2	Zend Studio.....	34
1.8.3	Aptana Studio.....	34
1.9	Justificación de la propuesta de solución. ....	35
1.10	Conclusiones del capítulo.....	36
Capítulo 2.....		37
2.	Introducción. ....	37
2.1	Características del sistema. ....	37
2.1.1	Modelo de Dominio .....	37
2.1.2	Conceptos del Modelo de Dominio.....	38
2.1.3	Especificación de Requisitos.....	38
2.1.4	Actores del Sistema .....	41
2.1.5	Diagrama de Casos de Uso del Sistema (DCUS).....	41
2.1.6	Descripción de Casos de Uso del Sistema.....	42
2.2	Análisis y Diseño .....	49
2.2.1	Patrón arquitectónico. ....	49
2.2.2	Patrones de diseño. ....	50
2.2.3	Clases del Diseño. ....	53
2.2.4	Modelo de Clases Persistentes. ....	58
2.3	Implementación.....	59
2.3.1	Diagrama de componentes. ....	59
2.3.2	Diagrama de Despliegue.....	60
2.3.3	Interfaz de la aplicación.....	60
2.4	Validación de la solución propuesta. ....	66
2.4.1	Métodos de Prueba.....	66
2.4.2	Pruebas aplicadas al software.....	67
2.5	Conclusiones.....	67
Conclusiones Generales .....		68

Recomendaciones.....	69
Referencias Bibliográficas .....	70
Glosario de Términos .....	72
Anexos .....	<b>¡Error! Marcador no definido.</b>

## Introducción

El desarrollo de la informática se encuentra presente en todas las esferas de la sociedad mundial. Las Tecnologías de la Información y las Comunicaciones (TIC) comprenden el conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de la información, en forma de voz, imágenes y datos, contenidos en señales de naturaleza acústica, óptica o electromagnética (1). Su objetivo fundamental es el uso y el acceso a la información, constituyendo un valor añadido al desarrollo de la humanidad.

Con el objetivo de lograr un mayor aprovechamiento de los recursos tecnológicos y debido a la acelerada evolución de Internet hacia portales con mayor cantidad de contenido y la alta participación de los usuarios directamente, a través de blogs y redes sociales, han dado lugar a los Sistemas de Gestión de Contenidos -CMS por sus siglas en inglés- siendo una herramienta esencial para todos los desarrolladores web, ya que posibilitan la gestión de los contenidos en línea. Los primeros CMS fueron desarrollados por organizaciones que publicaban una gran cantidad de contenido en Internet, y necesitaban de continuas actualizaciones; como revistas en línea, periódicos y publicaciones corporativas.

Cuba, a pesar de ser un país bloqueado económicamente, presta especial atención al sector de la educación, en el que se han realizado diversas transformaciones, donde las tecnologías desempeñan un papel protagónico. Actualmente en el sistema de enseñanza preescolar no se cuenta con la cantidad necesaria de software educativo para impartir los conocimientos suficientes y lograr un mayor aprendizaje en los niños. Con la idea de potenciar el desarrollo en materia de software surge La UCI, que tiene como objetivo fundamental, "...Convertir la Informática en una de las ramas más productivas y aportadoras de recursos para la nación. Es el empleo a fondo de la inteligencia y del capital humano que tenemos y principalmente del que podemos crear casi como espina dorsal de la economía" (2). En su afán de crecer como empresa productora de software, la UCI, cuenta con varios centros de desarrollo para elevar su producción y calidad, donde ha fortalecido el desarrollo de diversos productos, bajo las pautas de los CMS. Un ejemplo de esto es el proyecto "Producción de Recursos Didácticos", del Centro de Tecnologías para la Formación (FORTES) de la facultad 4, con el objetivo de potenciar la producción de materiales educativos que posibiliten la gestión de los contenidos. El producto "Mis Mejores Cuentos", es una multimedia educativa dirigida a niños de 3 a 5 años de edad, que tiene el propósito de digitalizar los cuentos infantiles más populares entre los niños de estas edades. Esta multimedia está estructurada por colecciones de 12 cuentos y cada uno de ellos contiene los módulos de narración, texto del cuento, sonidos, imágenes, autor asociado al cuento y entretenimientos, conformados por juegos como el dominó, rompecabezas, dibujo y construcción del cuento, con el propósito de desarrollar habilidades en los infantes de las edades antes mencionadas.

Todo el proceso de adicionar, modificar y eliminar un cuento determinado con todas sus medias (narración, imágenes, sonidos, texto y entretenimientos) en la multimedia "Mis Mejores Cuentos" los

**Herramienta de Administración de Cuentos del producto "Mis Mejores Cuentos".**

desarrolladores lo realizan manualmente y es un proceso muy engorroso y lento ya que se trabaja directamente con el código fuente del sistema. Por tanto el producto "Mis Mejores Cuentos" no tiene una herramienta de administración para el manejo de su contenido siendo de gran utilidad para el crecimiento del mismo.

Por lo anteriormente expuesto surge el siguiente **problema**: ¿Cómo administrar los recursos multimedia del software educativo "Mis Mejores Cuentos" sin necesidad de acceder al código fuente del producto?

El **objetivo general** para la presente investigación es desarrollar una Herramienta de Administración de Cuentos sobre plataforma web que permita gestionar los recursos multimedia del software educativo "Mis Mejores Cuentos".

Se define como **objeto de estudio** el proceso de desarrollo de herramientas administración web y gestión de contenidos en software educativo y el **campo de acción** está centrado en el desarrollo de la Herramienta de Administración de Cuentos para el software educativo "Mis Mejores Cuentos".

Se presenta como **Idea a defender** para la presente investigación: La implementación de la Herramienta de Administración de Cuentos para el producto "Mis Mejores Cuentos" permitirá gestionar las medias del producto "Mis Mejores Cuentos" sin necesidad de acceder al código fuente del mismo.

Para cumplir con el objetivo general planteado se definen los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación y definir la posición teórica del equipo de investigadores.
- ✓ Realizar el análisis y diseño de la Herramienta de Administración de Cuentos que posibilite la gestión de las medias del producto "Mis Mejores Cuentos".
- ✓ Implementar la Herramienta de Administración de Cuentos.
- ✓ Validar la propuesta de solución presentada.

Para dar cumplimiento a los objetivos específicos se definen las **tareas de investigación** a desarrollar:

- ✓ Selección y revisión bibliográfica.
- ✓ Procesamiento y evaluación de la información obtenida.
- ✓ Identificación de los requisitos del software a implementar, teniendo en cuenta las técnicas de ingeniería de requisitos.
- ✓ Diseño del sistema utilizando estereotipos web y aplicando patrones de diseño.
- ✓ Implementación de las clases, métodos y funciones de la Herramienta de Administración de Cuentos.
- ✓ Realización de las pruebas para comprobar la calidad del software desarrollado.
- ✓ Documentación de todo el proceso de desarrollo del sistema.

Herramienta de Administración de Cuentos del producto "Mis Mejores Cuentos".

La Herramienta de Administración a implementar está predefinida para ser utilizada en la multimedia web "Mis Mejores Cuentos", pero puede ser adaptada a cualquier software (SW) con características similares, o sea, una multimedia sobre plataforma web que contenga como recursos (imagen, texto, sonido, video) de cualquier área económica, política o social del país, en esto radica el **Impacto social** de la presente investigación.

Para el desarrollo de la investigación se pondrán en práctica una serie de métodos científicos para la obtención de la información y su procesamiento, así como para poder arribar a conclusiones. Dentro de los **métodos teóricos** a emplear se encuentran:

- ✓ El método de Análisis Histórico Lógico con el que se realizará un estudio crítico del producto para el cual se va a diseñar la Herramienta de Administración de Cuentos, así como de herramientas similares que se hayan desarrollado, para obtener un punto de referencia y comparar los resultados obtenidos.
- ✓ El método de Análisis y Síntesis permitirá llegar a ideas generales a partir de condiciones específicas, y obtener un resumen de un gran cúmulo de información después de haber sido analizada.
- ✓ A través del método de Modelado mediante la utilización del lenguaje UML permitirá reflejar la estructura, relaciones internas y características de la solución a través de diagramas, cuando se requiere modelar el dominio y especificar los artefactos a generar, se especificarán además los requisitos funcionales y no funcionales que el software debe cumplir.

También serán puestos en práctica **métodos empíricos** para obtener la percepción directa del objeto de investigación y del problema de la misma.

- ✓ El método de Entrevista se realizó a modo de reunión con los líderes y el equipo de desarrollo de la multimedia, haciendo preguntas con el objetivo de entender el funcionamiento e implementación de dicho sistema.
- ✓ El método de Encuesta se aplicó a modo de preguntas a profesores y estudiantes, para recopilar información necesaria que brinda una visión para fundamentar la problemática existente.

El desarrollo de la investigación del presente trabajo está compuesto por varias secciones que incluyen dos capítulos. Para lograr una mejor comprensión, de estas secciones a continuación se muestra un pequeño resumen de cada una de ellas:

- ✓ En el **Capítulo 1 “Fundamentación Teórica”** se realizará un estudio del arte sobre el objeto de estudio y campo de acción de la propuesta, así como la selección de las herramientas, los lenguajes de modelado y programación, framework de desarrollo y la metodología con las que se desarrollará el sistema.

- ✓ En el **Capítulo 2 “Propuesta de Solución”** se describen las características del sistema y los requisitos funcionales y no funcionales con los que debe cumplir, además se plantea el modelo de dominio donde se identifican los conceptos más importantes o los principales eventos que suceden en el entorno donde se encuentra el sistema. Se realiza una propuesta de solución a partir de los diferentes modelos y diagramas y se expone cómo está implementado el sistema en términos de componentes, además para comprobar la factibilidad del sistema se realizan diferentes pruebas al mismo.

# Capítulo 1

## FUNDAMENTACIÓN TEÓRICA

### 1. Introducción.

En el presente capítulo se abordan temas relacionados con el sistema que se propone, así como la definición de los conceptos vinculados al mismo, realizando un análisis crítico de las metodologías y tecnologías más usadas para el desarrollo de aplicaciones Web, así como de las herramientas y lenguajes para el modelado y desarrollo del sistema, seleccionando las que mejor se ajusten al tema abordado y justificando dicha propuesta.

### 1.2 Estado del arte.

La tecnología educativa, ha servido de apoyo para aumentar la efectividad del trabajo del profesor, sin llegar a sustituir su función educativa y humana, poniendo en práctica la utilización de medios didácticos y recursos educativos con una finalidad pedagógica para facilitar el proceso de Enseñanza-Aprendizaje. Logrando estimular la función de los sentidos para acceder de manera fácil a la adquisición de habilidades y convertirse en recursos indispensables para favorecer el aprendizaje en los estudiantes (3).

#### 1.2.1 Software Educativo.

El Software Educativo (SWE) es uno de los factores que ha representado un papel muy importante en el desarrollo del proceso Enseñanza-Aprendizaje. Constituye una evidencia del impacto de la tecnología en la formación de las personas, pues es una reciente herramienta didáctica, que se ha convertido en una alternativa válida para ofrecer al usuario un ambiente propicio para que fluya el conocimiento. La elaboración de estos sistemas tiene que ser un proceso fértil, de creatividad, con un fundamento pedagógico, y que realmente se obtenga con el efecto deseado (4). En el mundo existen aplicaciones educativas destinadas para todas las edades y para todo tipo de conocimientos, no solo se refieren a sitios educativos sino de cualquier índole. En Cuba se ha puesto a disposición de los estudiantes, computadoras y equipos, fomentando su uso como medio de enseñanza. Actualmente se desarrollan una serie de SWE que contribuyen a elevar el nivel del aprendizaje, unidos a estos se encuentran los CMS que se ponen en práctica cuando el sistema contiene un flujo de información de gran tamaño y variabilidad.

#### 1.2.2 Sistemas de Gestión de Contenidos (CMS).

En los últimos años se ha desarrollado el concepto de CMS, se trata de herramientas que permiten crear y mantener una web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las webs (5). El objetivo de estos CMS es facilitar la gestión de los contenidos en un sitio web, posibilitando el mantenimiento y la actualización por usuarios que no tengan una experiencia en el desarrollo de software, conjuntamente proporcionan al sistema una gran interactividad y eficiencia ya que todas las páginas se cargan dinámicamente, según

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

las peticiones del usuario, aunque estos beneficios que brindan no se limitan solo al entorno web. Por lo que se ha podido analizar estos sistemas ahorran mucho tiempo y recursos debiendo considerarse de alto costo, pero existen potentes herramientas de gestión de acceso libre con licencias de código abierto. Los CMS brindan una interfaz que controla una o varias bases de datos y permite manejar de forma independiente el contenido del diseño, siendo posible modificar el diseño en cualquier momento sin afectar o modificar el contenido existente y viceversa.

A inicios de los años noventa, el concepto de CMS era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación. En el año 1994 se emplea por primera vez una base de datos de objetos como repositorio de los contenidos de una web. RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos pero no fue hasta a finales del año 1995 que presentaron su CMS basado en una base de datos. Entre los CMS de código abierto uno de los primeros fue Typo 3, que empezó su desarrollo el año 1997, sin embargo, no es hasta el año 2000 que se comienza a consolidar esta idea con la aparición de la herramienta PHPNuke y su primera versión supuso tres semanas de trabajo al creador, rescribiendo el código de otra herramienta, Thatware.

Actualmente la tendencia es hacia la ampliación de las funcionalidades de los CMS, incorporando estándares que mejoran la compatibilidad de componentes, facilitando el aprendizaje al cambiar de sistema aportando calidad y estabilidad. Son muchas las ventajas que brindan los CMS dentro de las cuales cabe destacar la facilidad de incluir nuevas funcionalidades prácticamente sin costo, también es posible mantener un control de acceso al sitio, y todos los contenidos se encuentran centralizados en una base de datos (6).

En el contexto educativo los CMS más empleados son:

### ***Joomla***

Permite crear sitios web de alta interactividad, profesionalidad y eficiencia. Su administración está enteramente basada en la gestión en línea de contenidos (7). La interfaz administrativa es tan sencilla y amigable, que cualquier persona puede administrar sus propios contenidos web sin la necesidad de poseer conocimientos técnicos. Joomla no brinda mucho control sobre los visitantes al sitio y la asignación de recursos, tampoco ofrece alta seguridad ya que la información del sitio puede ser obtenida por terceros.

### ***Drupal***

Sistema de ayuda en línea tanto para usuarios como administradores. El código fuente de Drupal está libremente disponible bajo los términos de la licencia GPL, lo que posibilita adaptarlo según las necesidades (7). Los permisos son basados en roles, ya que los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles.

### ***MediaWiki***

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

Es uno de los CMS más populares que se especializan en wikis; fue desarrollado inicialmente para Wikipedia, pero luego fue hecho público bajo la Licencia GNU GPL (7). Cuenta con las herramientas básicas para la creación de contenidos en forma colaborativa. Soporta múltiples usuarios y diferentes niveles de acceso, posee la capacidad para manejar varios formatos de imagen. No existe control ni supervisión de la calidad de los contenidos, también se viola el derecho de autoría.

### **1.2.3 ¿Qué es la Gestión de Contenidos Educativos?**

La Gestión de Contenidos no son únicamente materiales o documentos de aprendizaje, sino también pueden ser elementos informativos o comunicativos que se encuentran en un entorno virtual. Los contenidos expuestos en los materiales configuran los contenidos educativos en entornos virtuales de aprendizaje (8).

Por tanto, para la presente investigación se define como todos los materiales puestos a disposición de alumnos o profesores en un entorno virtual de aprendizaje ya sea en tiempo real o referido, siendo de vital importancia en el nuevo mundo de las TIC para el desarrollo del proceso de Enseñanza-Aprendizaje.

### **1.2.4 ¿Qué es un LMS?**

Son Sistemas de Gestión de Aprendizaje (LMS) que posibilitan controlar el acceso a los contenidos, así como llevar un seguimiento de quienes utilizan la herramienta, facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenido académico permitiendo el seguimiento y la evaluación (9). Los LMS son diferentes de los CMS no solo por las características sino por el objetivo también, pero por razones de exigencia en el proceso educativo se ha hecho necesaria la unión de estos, obteniendo Herramientas de Gestión de Contenidos para el Aprendizaje (LCMS).

### **1.2.5 ¿Qué es un LCMS?**

Permiten la gestión de contenidos desde la creación de un objeto de aprendizaje hasta su publicación y seguimiento (9). Moodle constituye el LCMS más usado en la educación presencial y a distancia permitiendo gran flexibilidad para integrar contenidos, es capaz de alojar varios cursos, un ejemplo de esto es el Entorno Virtual de Aprendizaje (EVA), empleado en la UCI para fomentar el estudio y aprendizaje por parte de los estudiantes y profesores.

### **1.2.6 Herramientas de Administración Web.**

En la actualidad en el mundo se intensifica el uso de herramientas de administración para contenidos Web, estos pueden ser desarrollados tanto en software libre como propietario. Un sistema de administración de contenido es una herramienta que permite a compañías o individuos crear y modificar el contenido de su página Web, con poco o nada de conocimiento técnico. Todo esto bajo la tendencia e influencia del paradigma de desarrollo de una herramienta de administración de contenidos Web con el soporte de una comunidad de desarrolladores de extensiones y plantillas, como es el ejemplo de WordPress.

**Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.**

La mayoría de las herramientas de administración de contenido comparten algunas características básicas, como son, la capacidad para corregir y publicar contenido a través de un navegador, cambiar el diseño de las páginas usando plantillas y proporcionar el acceso a usuarios múltiples, pero algunos de estos sistemas son configurados a las necesidades particulares de cada cliente en específico.

Las herramientas de administración de contenido mayormente trabajan en conjunto con una base de datos que es utilizada para almacenar documentos, texto, imágenes y sonidos. Con frecuencia existe también un motor de flujo de trabajo, algo que permite distribuir las páginas Web para revisión antes de su publicación. Un sistema de administración de contenido es en sí una interfaz a una base de datos que permite a los usuarios autorizados editar y publicar nuevo contenido.

Es así que el personal responsable de un sitio Web, puede agregar contenido al sitio sin tener que escribir una sola línea de código. Estas herramientas en la mayoría de los casos se operan a través de navegadores web, donde la información puede ser accedida desde cualquier computadora. Lo más sobresaliente de todo esto es que los sistemas de administración de contenido Web generan automáticamente las nuevas páginas y sub-niveles de navegación.

A la hora de administrar alguna aplicación Web es mucho más cómodo hacerlo mediante los CMS debido a que son muy eficientes para llevar el control y administración de los contenidos y realizan algunas funciones muy semejantes a una herramienta de administración Web. La mayoría de los sitios y portales Web de la intranet cubana son realizados con CMS como herramienta de administración y un ejemplo de esto es el portal de cultura de la provincia de Matanzas<sup>1</sup> que utiliza el CMS Drupal.

En la UCI se ha puesto en práctica la utilización de CMS como herramientas administrativas de contenidos en la Web y como ejemplo de esto se tiene el portal "Octavitos"<sup>2</sup> de la Facultad 4 aunque también existen algunas tendencias al desarrollo e implementación de nuevas herramientas de administración de páginas Web.

Los gestores de contenido, se pueden clasificar según diferentes criterios, entre los que se encuentran:

✓ **Por sus características:**

- Según el lenguaje de programación empleado, como por ejemplo: ASP, Java, PHP, ASP.NET, Ruby On Rails, Python.
- Según la licencia: Código abierto o software privativo.

✓ **Por su uso y funcionalidad:**

- Blogs: CMS para páginas personales.
- Foros: CMS para compartir opiniones.

---

<sup>1</sup> Este portal se encuentra disponible en: <http://www.atenas.cult.cu>

<sup>2</sup> Este portal se encuentra disponible en: <http://octavitos.uci.cu>

- Wikis: CMS para el desarrollo colaborativo.
- Enseñanza: Plataforma para contenidos de enseñanza on-line.
- Comercio electrónico: Plataforma de gestión de usuarios, catálogo, compras y pagos.
- Publicaciones digitales: CMS para realizar publicaciones on-line.
- Difusión de contenido multimedia: CMS para la publicación de textos, imágenes, sonidos, entre otros.

Basado en lo expuesto anteriormente el sistema a desarrollar se clasifica en gestor de difusión de contenido multimedia.

### 1.2.7 Herramientas Cubanas.

En Cuba, se han elaborado dos herramientas para el desarrollo del software educativo cubano, que se nombran HAEduc y SadHEA Web, y se han creado también componentes en Joomla, para poder migrar las colecciones de sistemas educativos desarrolladas hasta el momento (10).

#### **SadHEA**



**Fig. 1 Herramienta Sadhea.**

SADHEA-Web (Sistema de Autor para el desarrollo de hiperentornos de aprendizaje para la web), es una solución 100% cubana para la producción de software educativo, desarrollado en PHP, MySQL y Javascript, con tecnología AJAX, que ofrece grandes beneficios a los docentes que se interesen por elaborar su propio Software Educativo. Es un Sistema de Autor para el desarrollo de Hiperentornos de Aprendizajes para la Web, desarrollado por el Centro de Estudios de Software y sus Aplicaciones Docentes (CESOFTAD) (10).

El sistema satisface las demandas actuales para transferir el software educativo cubano a multiplataforma, como parte de los esfuerzos del país por el logro de la independencia tecnológica. Además, reduce el tiempo de montaje (etapa de programación) de los SWE a desarrollar bajo esta concepción, elevando la eficiencia del proceso y minimizando los errores típicos de esta fase de desarrollo.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

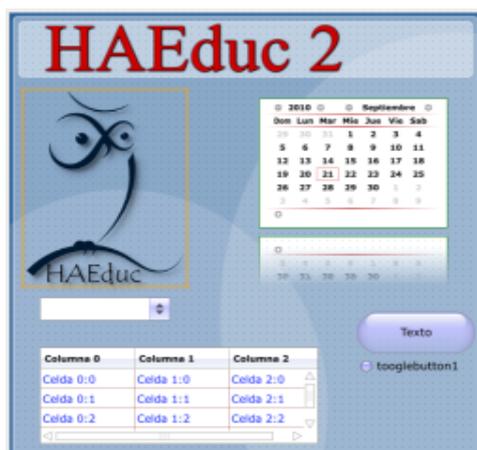


Fig. 2 Herramienta Haeduc.

HAEduc es una herramienta de autor multiplataforma, que surge producto de la necesidad de herramientas libres para elaborar recursos educativos en los ambientes Windows y Linux. Desarrollada por el Centro de Estudios de Software Educativos (CENSOE), está destinada al desarrollo de recursos educativos informáticos, constituye un diseño único de su tipo en Cuba para la gestión de aplicaciones. Permite la creación de recursos mucho más sofisticados, dinámicos y altamente efectivos por los niveles de interactividad que pueden ser alcanzados, como multimedia, colecciones de software educativos, entre otras (10). HAEduc es capaz de generar los productos de forma cruzada, por ejemplo, desde Windows es posible generar el producto para que se ejecute en Linux, y viceversa.

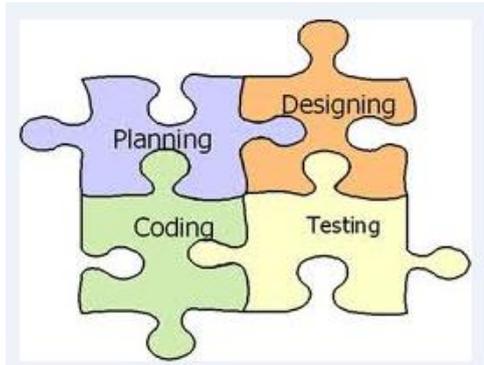
La idea principal es crear un software mediante el cual una vez instalado el producto se pueda gestionar todo el contenido de la medias que pertenezcan a este sistema “Mis Mejores Cuentos” teniendo en cuenta medidas de seguridad para que los cambios realizados solo sean ejecutados por personal autorizado para hacerlo.

### 1.3 Metodologías de desarrollo.

Las metodologías de desarrollo de software son un conjunto de procedimientos y técnicas que permiten conocer los elementos necesarios para desarrollar y documentar productos de software. Se dividen en dos grandes grupos, las metodologías ágiles, en las que se interactúa constantemente con el cliente, son usadas para proyectos pequeños y de corta duración, y las metodologías tradicionales que son empleadas en grandes proyectos donde se requiera una gran cantidad de documentación durante todo el ciclo de vida del mismo y en muchas ocasiones no se tiene un contacto directo con el cliente. “Para desarrollar un proyecto de software es necesario establecer un enfoque disciplinado y sistemático. Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida” (11). Seguidamente se hace una valoración de las metodologías más usadas, siendo válido destacar que “...ninguna metodología hace

el trabajo por sí sola, pero te podrá ayudar” (12) en todo lo referente a la organización y control del trabajo que se realiza.

### 1.3.1 Programación Extrema (XP).



**Fig. 3 Fases de la metodología XP.**

XP nace buscando simplificar el desarrollo del software y reducir el costo del proyecto. Está enfocada hacia la satisfacción del cliente y se emplea en proyectos pequeños y con una amplia rotación de roles, por lo que es muy exitosa en la actualidad. Reduce el costo del cambio en las etapas de vida del sistema. Es una programación rápida en la que clientes, desarrolladores y administradores conforman un equipo dedicado a la producción de software de calidad ya que XP implementa una forma simple pero muy efectiva en el trabajo como equipo. Rediseña el código para dejarlo en el estado más simple posible (13).

Esta metodología permite al equipo definir sus propias reglas a seguir, que pueden cambiar en cualquier momento. En XP se piensa que no existe un proceso que sirva para todos los proyectos; lo que se hace habitualmente es adaptar un conjunto de prácticas simples a las características de cada proyecto. Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

La programación del software siempre se define en pareja con el objetivo principal de lograr mayores resultados y los menores errores posibles, mientras uno codifica haciendo énfasis en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado.

En XP se definen Historias de Usuario como base del software y estas historias las escribe el cliente describiendo escenarios sobre el funcionamiento del sistema a desarrollar. Presenta un diseño evolutivo donde no tiene importancia el análisis como fase independiente debido a que se trabaja en función de las necesidades del momento (13).

XP necesita un alto esfuerzo de desarrollo y es posible que por retrasos el prototipo se convierta en parte del producto, estas son dos de las desventajas que presenta esta metodología.

Esta metodología es muy buena, pero una de sus características es que cuenta con el cliente al lado del equipo de desarrollo y en este caso no se refleja de esa manera.

### 1.3.2 SCRUM

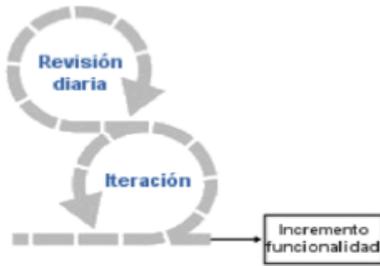


Fig. 4 Estructura de SCRUM.

Define un marco para la gestión de proyectos, está pensada para soportar un acelerado cambio de requisitos, el desarrollo del software es realizado mediante iteraciones o sprint con una duración de 30 días y el resultado de cada uno de ellos constituye una entrega parcial del producto final que se le muestra al cliente. Durante el proyecto se realizan reuniones pudiendo ser diaria la secuencia de las mismas consolidando la integración y coordinación del equipo de trabajo (14). SCRUM también se emplea para resolver situaciones en las que se ha demorado demasiado la entrega de un producto, cuando no se está entregando al cliente lo que este necesita o los costos se elevan, aunque se relaciona muy bien con XP no define que hacer para el código por lo que es recomendable complementar su uso con otra metodología de desarrollo.

Scrum controla de forma efectiva y adaptable la evolución del proyecto, empleando las siguientes prácticas (14):

- ✓ Revisión de las iteraciones.
- ✓ Desarrollo incremental.
- ✓ Desarrollo evolutivo.
- ✓ Auto-Organización.
- ✓ Colaboración.

Algunas de las desventajas de SCRUM es que requiere delegar responsabilidades al equipo. Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas, por lo que no resulta factible aplicarla en el proceso de desarrollo del sistema que se propone.

### 1.3.3 Microsoft Solutions Framework (MSF).

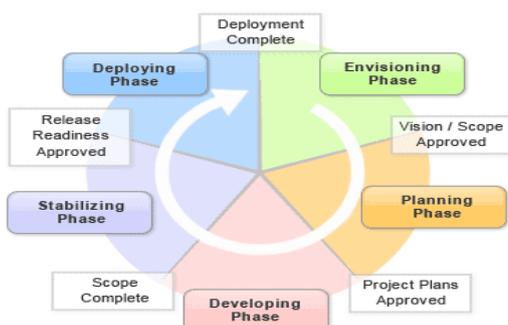


Fig. 5 Ciclo de Desarrollo de la Metodología MSF.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

MSF se centra en dar una solución exitosa, rápida y con la mayor reducción posible de personas y riesgos, es adaptable, flexible y escalable, ha evolucionado con la experiencia de los grupos de trabajo obteniendo un framework más fácil de entender y adoptar.

Existen dos modelos de MSF, uno de ciclo de vida de desarrollo de software descriptivo en el que se documenta el proceso pasivamente siendo muy útil como base de conocimiento y mejora de procesos de desarrollo de software, mientras que un modelo prescriptivo describe el proceso en término de los jugadores involucrados, la secuencia de actividades, y el producto final.

MSF se enfoca en promover comunicaciones abiertas, trabajar para una visión compartida fortaleciendo los miembros del equipo, estableciendo responsabilidades claras y compartidas, aprendiendo de las experiencias anteriores.

Como principales desventajas de MSF se puede mencionar que el análisis de riesgo suele llevar mucho tiempo frenando el avance del proyecto y al ser un modelo de Microsoft implica que se tiene que utilizar herramientas solo de Microsoft.

#### **1.3.4 Metodología de Administración de Relaciones (RMM)**

RMM se define como un proceso de análisis y diseño, basado en conceptos del Modelo de diseño de Hipertexto (HDM) y desarrollo de aplicaciones de estructura estable. Se centra en el diseño, desarrollo y fases de construcción. Impone una disciplina de trabajo sobre el proceso de desarrollo de software, con el objetivo de asegurar la obtención de una aplicación que satisfaga los requisitos del usuario y reúna estándares aceptables de calidad.

RMM constituye una metodología tentadora para el desarrollo del proceso por el desglose de las fases de la producción y la incorporación de diagramas para el diseño de la presentación, el comportamiento dinámico y la estructura de la navegación.

“Es una metodología que asume un ciclo de vida completo, compuesto de siete etapas, en las que se va modelando la estructura de la aplicación, antes de estas etapas se determinan objetivos, se analizan necesidades y tipos de navegación.” (15).

Al final del desarrollo del software y después de haber pasado por las diferentes etapas se realizan pruebas en busca de fallas y errores con los que se llevan a cabo evaluaciones experimentales para corregir dichas inconformidades encontradas.

Esta metodología a pesar de sus ventajas no se ajusta adecuadamente a las necesidades del proyecto a realizar. Además habría que efectuar un estudio para aumentar el conocimiento sobre esta metodología y el equipo de desarrollo no cuenta con el tiempo requerido para esta actividad.

### 1.3.5 RUP (Rational Unified Process)

RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, emplea una forma organizada de asignar tareas y responsabilidades, o sea define quién hace qué, cuándo y cómo.

RUP describe actividades y artefactos que se deben obtener en el proceso de desarrollo del software. Los procesos de RUP estiman tareas y horario del plan midiendo la velocidad de las iteraciones. Se basa en las mejores prácticas y describe cuatro elementos fundamentales (16):

- ✓ Trabajadores (¿Quién?).
- ✓ Actividades (¿Cómo?).
- ✓ Artefactos (¿Qué?).
- ✓ Flujo de Actividades (¿Cuándo?).

Define 9 flujos de trabajo a realizar en cada fase del proyecto son, los 6 primeros conocidos como flujos de ingeniería (Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Prueba y Despliegue), y los otros tres como flujos de apoyo (Administración de Proyecto, Administración de Configuración y Cambios y Ambientes.)

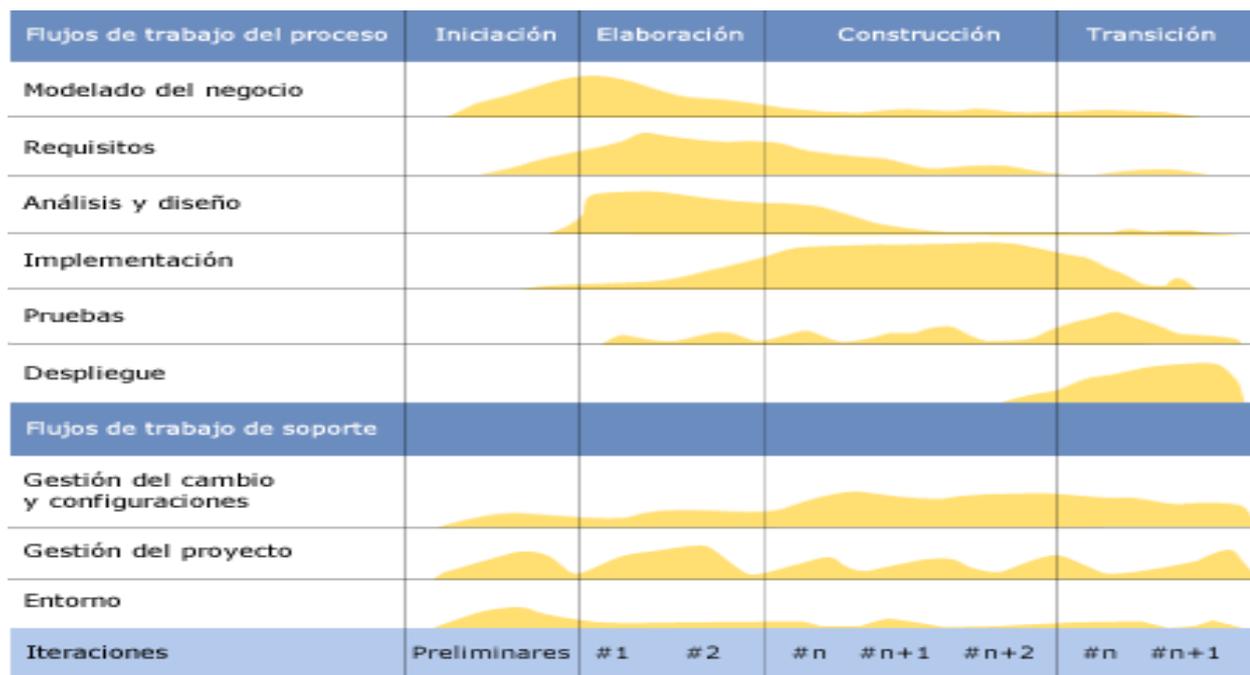
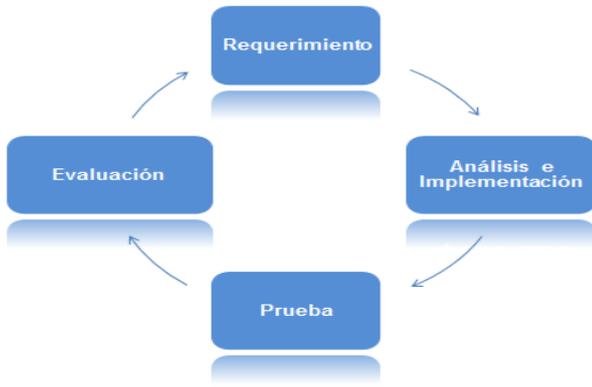


Fig. 6 Ciclo de vida de RUP.

El proceso de desarrollo del RUP está dividido en ciclos e iteraciones obteniéndose un producto final como resultado. Cada ciclo se divide en 4 fases: fase de inicio, elaboración, construcción y transición. Es una metodología de desarrollo iterativa e incremental y cada uno de los 9 flujos se puede descomponer en iteraciones y en cada iteración se realiza una entrega de producto ejecutable.



**Fig. 7 RUP como proceso iterativo e incremental.**

Sus tres características principales son (16):

- ✓ Dirigido por casos de uso.
- ✓ Iterativo e incremental.
- ✓ Centrado en la arquitectura.

RUP lleva a cabo un profundo levantamiento de requisitos, buscando detectar defectos en las fases iniciales con el objetivo de reducir tanto como sea posible el número de cambios en el producto que se espera obtener (16). Realiza el análisis y diseño lo más completo posible intentando anticiparse a futuras necesidades, a diferencia de XP en RUP el cliente interactúa con el equipo de desarrollo mediante reuniones.

Esta metodología es la que más se ajusta al desarrollo de la aplicación debido a que el proyecto "Producción de Recursos Didácticos" al que se integrara la Herramienta de Administración de Cuentos, es un proyecto a largo plazo que cuenta con 3 años de vida y es necesario una buena documentación como la que propone la metodología RUP, además en la arquitectura del proyecto se define el uso de esta metodología de desarrollo.

#### **1.4 Lenguajes de modelado.**

El modelado de sistemas software es una técnica para tratar con la complejidad de estos sistemas. El uso de modelos ayuda al ingeniero de software a visualizar el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente.

##### **1.4.1 Lenguaje Unificado de modelado (UML).**

UML es un estándar muy usado en la industria del software ya que permite visualizar, especificar, construir y documentar los elementos que forman un sistema orientado a objetos. Fue creado para posibilitar el intercambio de modelos entre herramientas CASE (Ingeniería de Software Asistida por Computadora) orientadas a objetos (17).

UML está pensado para sistemas complejos debido a que mediante un plano -visión global- es mucho más fácil detectar las dependencias y dificultades del sistema y los cambios en una etapa inicial resultan más fáciles de realizar que en una etapa final, su paradigma está basado en la orientación a objetos por lo que para trabajar con este lenguaje es necesario tener conocimiento de este paradigma.

Herramienta de Administración de Cuentos del producto "Mis Mejores Cuentos".

Con el objetivo de hacer el lenguaje más extensible y permitir la validación y ejecución de los modelos fue creada la versión 2.0 de UML, permitiendo modelar todo tipo de sistemas (17).

Entre las Ventajas que ofrece este lenguaje encontramos, que se puede emplear tanto en sistemas de hardware como de software, a través de diagramas establecidos (17). Es una consolidación en cuanto a orientación a objetos se refiere. Está apoyado como la notación estándar para desarrollar proyectos informáticos y además promueve la reutilización.

Debido a las características expuestas anteriormente, UML es el lenguaje de modelado que se selecciona para el desarrollo del sistema.

#### **1.4.2 Lenguaje para modelar Aplicaciones Multimedia (OMMMA-L)**

Facilita el modelado de aplicaciones multimedia interactivas, incorpora las clases correspondientes a las medias, divide en dos áreas el diagrama de clases, una para la jerarquía y otra para la estructura lógica.

OMMMA-L soporta el modelado de aspectos estructurales y funcionales de un sistema y su interfaz de usuario, centrándose en la funcionalidad desde la perspectiva del sistema de software, y su sintaxis es definida explícitamente. Consta de cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular: vista lógica, vista de presentación espacial, vista de comportamiento temporal predefinido y vista de control interactivo (18).

OMMMA-L no es un lenguaje nuevo, sino una extensión del UML que permite modelar aspectos estructurales, funcionales y dinámicos de un sistema interactivo y su interfaz de usuario.

#### **1.4.3 Lenguaje para Modelar Aplicaciones Educativas (ApEM-L).**

ApEM-L es un lenguaje de modelado para ser empleado en aplicaciones educativas fundamentalmente, es una extensión de UML que permite incorporar elementos del proceso productivo UCI, también se incorporan los elementos más significativos de OMMMA-L, obteniéndose una extensión para modelar aplicaciones educativas. Este lenguaje no modifica la semántica de UML, sino que trabaja en estereotipos restrictivos que producen modificaciones descriptivas y decorativas en la representación de lenguaje base. Puede utilizar para su representación todas las herramientas CASE que existen actualmente.

ApEM-L incorpora la vista de presentación como aporte fundamental al modelado de este tipo de aplicaciones con dos diagramas que permiten definir la estructura para la navegación y la estructura para la presentación (19).

Debido a que el sistema que se propone no constituye una multimedia educativa que es para lo que está destinado especialmente este lenguaje no se selecciona para el desarrollo de este sistema.

## 1.5 Herramientas de ingeniería asistida por computadoras (Computer Aided Software Engineering (CASE))

En la actualidad el desarrollo de software se hace más complejo cada día, por lo que es preciso llevar una buena documentación de todo lo relacionado con el proceso de desarrollo. El conocimiento y buen manejo de las herramientas que nos permiten un mejor entendimiento del software que estamos desarrollando es muy importante ya que esto hace más portable la documentación a generar, proporciona la reutilización de componentes y permite generar automáticamente el código base, para esto se utilizan las herramientas case.

### 1.5.1 Visual Paradigm (VP)

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software entre los cuales se encuentran, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. VP es un producto de alta calidad que soporta aplicaciones web y varios idiomas, contiene generación de código para Java y exportación como HTML, es fácil de instalar y actualizar y es compatible entre ediciones.

Ayuda a una rápida construcción de aplicaciones con una mayor calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm para UML es una herramienta que soporta versiones de este lenguaje así como la notación y modelado de procesos de negocio, siendo un producto de excelencia porque facilita la organización de diagramas visuales y de diseño de los proyectos. Ayuda a desplegar al máximo el proceso de desarrollo de software. Emplea una rápida respuesta con poca memoria utilizando los tiempos del procesador, permitiendo manejar grandes y complicadas estructuras de un proyecto de forma eficiente.

#### Características:

- ✓ Soporte de UML.
- ✓ Diagramas de flujo de datos.
- ✓ Generación de bases de datos
- ✓ Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Importación y exportación de ficheros XML.
- ✓ Generación de código en varios lenguajes de programación como son: Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python.

Uno de los elementos a destacar es que Visual Paradigm no es una herramienta libre pero la UCI tiene la licencia para poder usar su versión 3.4, por tanto es una herramienta muy importante para el modelado y desarrollo de la herramienta propuesta.

### 1.5.2 Rational Rose Enterprise

Es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte UML. Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente (20).

#### Características

- ✓ Herramienta propietaria.
- ✓ La generación de código Ada, ANSI C ++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ✓ Incorpora un modelado Web que provee visualización, para representar aplicaciones Web.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requisitos de aplicación a través de diseños lógicos y físicos.
- ✓ Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- ✓ Capacidad para integrarse con cualquier sistema de control de versiones.

Esta herramienta a pesar de sus características y ventajas no es conveniente emplearla porque es una herramienta propietaria y el equipo de desarrollo no cuenta con la licencia para su utilización.

### 1.5.3 Umbrello UML Modeller

Es una herramienta que ayuda con todo lo referido a la documentación y diseño de diagramas para el modelado de una aplicación en un proceso de desarrollo de software. Es una herramienta libre, de código abierto y actualmente Umbrello permite instalarse en diferentes plataformas y posee más de treinta idiomas diferentes, gracias a su licencia original GPL. Según el manual de esta herramienta en el año 2001, se definen las siguientes características (21).

#### Características

- ✓ Umbrello UML Modeller incluye soporte para los siguientes lenguajes: C++, Java, C#, PHP, Javascript, Actionscript, SQL, Pascal, Python, Perl entre otros.
- ✓ Posee Interfaz Gráfica con diferentes áreas de trabajo.
- ✓ Se puede generar código fuente en varios lenguajes de programación como C++, CORBA IDL, Ada, Java y Python, a partir de la maqueta UML para ayudar a comenzar la implementación de su proyecto.

Es válido destacar que esta herramienta a pesar de ser de código abierto y de tener características favorables, no es conveniente su uso para el desarrollo de la herramienta en cuestión ya que el equipo de desarrollo no tiene los conocimientos necesarios para su uso y tampoco dispone del tiempo requerido para su estudio.

## 1.6 Lenguajes de desarrollo web.

### Lenguajes del lado del Cliente:

#### 1.6.1 Lenguaje HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web. Desarrollado por el World Wide Web Consortium. Es un lenguaje muy utilizado en la actualidad por todos los desarrolladores Web (22).

#### Ventajas:

- ✓ Sencillo que permite describir hipertexto.
- ✓ Texto presentado de forma estructurada y agradable.
- ✓ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- ✓ Archivos pequeños.
- ✓ Despliegue rápido.
- ✓ Lenguaje de fácil aprendizaje.
- ✓ Lo admiten todos los exploradores.

#### Desventajas:

- ✓ Lenguaje estático.
- ✓ La interpretación de cada navegador puede ser diferente.
- ✓ Guarda gran cantidad de etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- ✓ Las etiquetas son muy limitadas.

Este es un lenguaje muy importante y a pesar de sus desventajas es un lenguaje estándar que hay que incluir para la confección de las páginas web a desarrollar, se selecciona para el desarrollo de la Herramienta de Administración de Cuentos debido a las ventajas que ofrece.

#### 1.6.2 Lenguaje Javascript

Este es un lenguaje interpretado, no requiere compilación. Utilizado principalmente en páginas web. Es bastante similar a Java, aunque no es un lenguaje orientado a objetos permite el trabajo con clases, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript y puede ser integrado dentro de nuestras páginas web (23).

#### Ventajas:

- ✓ Lenguaje de scripting seguro y fiable.
- ✓ Los script tienen capacidades limitadas, por razones de seguridad.
- ✓ El código Javascript se ejecuta en el cliente.

#### Desventajas:

- ✓ Código visible por cualquier usuario.
- ✓ El código debe descargarse completamente.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

- ✓ Puede poner en riesgo la seguridad del sitio.

Se selecciona este lenguaje que a pesar de sus desventajas es muy importante para el equipo de desarrollo ya que es dinámico y permite implementar una serie de funciones importantes para el desarrollo de aplicaciones en la Web. Además sigue la línea a la cual va dirigida la implementación del software a desarrollar.

### 1.6.3 Lenguaje Python

Es un lenguaje de programación creado en el año 1990, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web. Su código no necesita ser compilado, por lo que se dice que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten un estilo de programación particular (24).

#### **Ventajas:**

- ✓ Gran cantidad de funciones y librerías.
- ✓ Sencillo y rápido de programar.
- ✓ Multiplataforma.
- ✓ Licencia de código abierto (Open Source).
- ✓ Orientado a Objetos.
- ✓ Portable.

#### **Desventajas:**

- ✓ Lentitud por ser un lenguaje interpretado.

Este lenguaje es muy usado por muchos programadores en la Web y es un lenguaje que tiene muchas ventajas con respecto a otros lenguajes y presenta muchas funcionalidades importantes para el desarrollo de aplicaciones pero la herramienta a desarrollar debe seguir la línea de programación especificada para el producto "Mis Mejores Cuentos" al cual será integrada.

#### **Lenguajes del lado del Servidor:**

### 1.6.4 Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitios web. PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión .php (25).

**Ventajas:**

- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objeto, clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

**Desventajas:**

- ✓ Se necesita instalar un servidor web.
- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

**Seguridad:**

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

Este lenguaje se selecciona para usarlo como tecnología del lado del servidor teniendo en cuenta que es un lenguaje muy estable y bastante seguro, además de tener todas las cualidades y funciones necesarias para un buen manejo de los datos en el servidor. Es un lenguaje muy usado en nuestra universidad y del cual se tiene una gran experiencia en su uso y una abundante documentación. También es muy importante destacar que es un lenguaje libre.

### 1.6.5 Lenguaje ASP

Es una tecnología del lado de servidor desarrollada por *Microsoft* para el desarrollo de sitios web dinámicos. Fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje necesitan tener instalado Internet Information Server (IIS). ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl. El código ASP puede ser insertado junto con el código HTML (26). Los archivos cuentan con la extensión (asp).

**Ventajas:**

- ✓ Usa Visual Basic Script, siendo fácil para los usuarios.
- ✓ Comunicación óptima con SQL Server.
- ✓ Soporta el lenguaje JScript (Javascript de Microsoft).

**Desventajas:**

- ✓ Código desorganizado.
- ✓ Se necesita escribir mucho código para realizar funciones sencillas.
- ✓ Tecnología propietaria.
- ✓ Hospedaje de sitios web costosos.

Este lenguaje no es muy adecuado para el desarrollo de la herramienta en cuestión ya que el equipo de desarrollo no cuenta con los recursos y tecnologías necesarios para la utilización de este lenguaje. Tampoco se cuenta con los conocimientos y la experiencia necesaria para su utilización ni con el tiempo requerido para el estudio de este lenguaje (26).

## 1.7 Framework de desarrollo.

Actualmente ya hemos pasado de páginas estáticas basadas solamente en HTML y CSS a páginas dinámicas que utilizan motores externos como PHP y ASP para dar este dinamismo. Aunque javascript no es una tecnología reciente los framework desarrollados con la misma han tomado mayor popularidad ya que le hacen el trabajo al desarrollador mucho más fácil debido a las múltiples funcionalidades que brindan.

### 1.7.1 Symfony

Está diseñado para optimizar el desarrollo de las aplicaciones Web, separando la lógica del negocio, la lógica del Servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Symfony le permite al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una aplicación (27).

**Características:**

- ✓ Implementa las reglas del negocio.
- ✓ Abstrae el acceso a datos.
- ✓ Es maduro, estable, profesional e infinitamente escalable.
- ✓ Simplifica el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes.
- ✓ Además proporciona estructura al código fuente, forzando al programador a crear códigos más legibles y más fácil de mantener.

### 1.7.2 ExtJs

ExtJs según un artículo publicado (28), incluye en sus funcionalidades las librerías de los framework JQuery, Prototype y Scriptaculous. Posee controles para Campos de Textos, incluyendo áreas de Texto, controladores, selectores de fecha, campos numéricos, para radiobox y checkbox. También componentes para crear y manipular DataGrids donde goza de cierta ventaja sobre otros frameworks. Es posible crear “ventanas” con Barras de Herramientas y los menús con estilo de aplicaciones de escritorio, diálogos modales y eventos. *ExtJs* puede ser adquirido bajo licencias Libres.

### 1.7.3 JQuery:

Es una librería liviana que enfatiza la interacción entre Javascript y HTML. Contiene selectores de elementos DOM y permite la modificación del mismo. Entre quienes utilizan jQuery están: Google, Dell, Sun, MLB, digg, NBC, CBS, Netflix, Technorati, mozilla.org, WordPress y Drupal. Con facilidad podría ser considerado el framework más utilizado en la actualidad. En adición, en internet existen bastantes plugins que agregan funcionalidades extras al framework. Actualmente su última versión estable, la 1.3.2, viene un solo archivo de 19KB, y su funcionalidad puede ser extendida utilizando plugins. Por todas las características y ventajas antes mencionadas se selecciona este framework para el desarrollo de la Herramienta de Administración de Cuentos.

### 1.7.4 Prototype

También se emplea Prototype ya que es una simple implementación de un solo archivo de código en Javascript que provee un framework para *Ajax* y otras herramientas. Contiene varias funciones para programar en Javascript que van desde accesos directos a funciones, elementos y objetos Javascript, hasta funciones para lidiar con XMLHttpRequest.

### 1.7.5 Scriptaculous

Scriptaculous es una librería javascript basada en Prototype que agrega efectos visuales dinámicos y una interface para elementos a través de DOM. Viene incluido en Seaside y Ruby on Rails. Entre quienes usan Scriptaculous están: NASA, Apple, CNN, Basecamp y Feedburner.

## 1.8 Entorno de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse exclusivamente a un sólo lenguaje de programación o bien, puede utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word (29).

### 1.8.1 NetBeans

Es una herramienta pensada para escribir, compilar, depurar y ejecutar programas. Es un IDE de código abierto escrito completamente en Java. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web y aplicaciones móviles). También tiene soporte para varios lenguajes como PHP, Javascript, HTML, CSS y en su versión 6.8 tienen soporte para el framework Symfony. Es un producto libre y gratuito sin restricciones de uso (30).

### 1.8.2 Zend Studio

Es un programa de la casa Zend, uno de los mayores impulsores de PHP, orientada a desarrollar aplicaciones web. Zend es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código, está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux (31).

#### Características:

- ✓ Soporte para PHP 4 y PHP 5.
- ✓ Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clases.
- ✓ phpDoc integrado.
- ✓ Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- ✓ Inserción automática de paréntesis y corchetes de cierre.
- ✓ Sangrado automático y otras ayudas de formato de código.
- ✓ Emparejamiento (matching) de paréntesis y corchetes (si se sitúa el cursor sobre un paréntesis (corchete) de apertura (cierre), Zend Studio localiza el correspondiente paréntesis (corchete) de cierre (apertura)).
- ✓ Detección de errores de sintaxis en tiempo real.
- ✓ Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP.
- ✓ Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- ✓ Soporte para gestión de grandes proyectos de desarrollo.
- ✓ Manual de PHP integrado.
- ✓ Soporte para control de versiones usando Subversion.
- ✓ Cliente FTP integrado.
- ✓ Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.

### 1.8.3 Aptana Studio

Aptana Studio es un IDE para la elaboración de aplicaciones web dinámicas que empleen PHP, Ruby, Ruby on Rails y Python. Su integración con dos nuevas herramientas de Aptana amplían sus

posibilidades casi al infinito. Estas son Aptana Jaxer, un servidor AJAX, y Aptana Cloud, un servicio de hosting complementario. Aptana Studio posee un asistente de código que ayuda al programador en la escritura de los diferentes lenguajes, CSS y Javascript entre ellos. En el caso del HTML, puede mostrar todos los elementos pertenecientes a este lenguaje y sus propiedades, también contiene información de soporte para los principales navegadores web: IE, Firefox, Opera, Netscape y Safari. Además se conoce a Aptana Studio como el sustituto de Dreamweaver para Linux (32).

### **Características:**

- ✓ Explorador de código en forma de árbol.
- ✓ Librerías populares AJAX/Javascript.
- ✓ Extensión de funcionalidad mediante macros y acciones.
- ✓ Visor de errores y advertencias.
- ✓ Servidor local para probar el código.
- ✓ Es gratis, de código abierto y basado en la plataforma Eclipse.
- ✓ Funciona en Windows, Mac y Linux.
- ✓ Visualización de errores de sintaxis a medida que se escribe.
- ✓ Debug en Firefox
- ✓ Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- ✓ Previsualización de estilos CSS con el editor CSS.
- ✓ Extensible a partir de plugins que puede crear tanto Aptana como otras empresas.

### **1.9 Justificación de la propuesta de solución.**

Para la propuesta de solución se selecciona como metodología de desarrollo RUP, mediante la que se obtiene una amplia documentación acerca del software, elemento imprescindible para un proyecto de larga duración y sobre todo si se desea comercializar, permite también la generación de una serie de artefactos descritos por esta metodología, además se logra un desarrollo centrado en una arquitectura sólida, ayuda a obtener un producto final con una mejor calidad, a través del control de riesgos y la aplicación de pruebas. No se propone una metodología ágil puesto que estas generan muy poca documentación, además el lenguaje que emplea RUP para el modelado es UML que especifica y documenta los artefactos a obtener a lo largo del desarrollo del producto, y como herramienta CASE se escoge Visual Paradigm que a pesar de no ser una herramienta libre la UCI cuenta con la licencia de la versión 3.4. Después del análisis realizado sobre los lenguajes de desarrollo web se seleccionan por las ventajas que brindan, el lenguaje de etiquetado HTML, como lenguaje del lado del cliente Javascript y del lado del servidor el lenguaje PHP y como framework de desarrollo fue seleccionado Prototype y JQuery debido a las disímiles ventajas que brinda y lo diferencian de otros framework y como IDE de desarrollo se seleccionan Zend Studio y Aptana Studio, ya que se integran perfectamente con los lenguajes seleccionados para el desarrollo del software y son muy utilizados a nivel mundial por los desarrolladores web.

### **1.10 Conclusiones del capítulo.**

En este capítulo fueron abarcados los principales elementos del estado del arte, explicando conceptos de importancia ligados al desarrollo de la informática en la rama de la educación a nivel mundial y nacional. Mediante la realización del estudio y análisis de las características de algunas herramientas, metodologías y lenguajes de modelado y desarrollo web se seleccionaron los más convenientes para la implementación del sistema en cuestión. También se realizó un análisis de las tendencias de los CMS unidos al SWE en el proceso de Enseñanza-Aprendizaje a nivel global y nacional y se expuso una reseña del Framework que se escogió para ser utilizado.

### 2. Introducción.

En el presente capítulo se describen los conceptos del dominio, se identifican roles y casos de usos que intervienen en el modelado del sistema para la conformación de la descripción de la propuesta de solución. También se especifican los requisitos funcionales y no funcionales del sistema propuesto, y se identifican los casos de usos, actores y las relaciones existentes entre ellos mediante el modelo de casos de uso del sistema. Además se proponen el modelo de análisis con los diagramas de clases del análisis, así como el modelo de diseño con sus diagramas de clases del diseño, describiendo como son implementados los elementos de este modelo y la forma en que va a estar distribuido el sistema a través de los diagramas de componentes y despliegue, y para comprobar el funcionamiento de la herramienta propuesta se realizan una serie de pruebas al software.

#### 2.1 Características del sistema.

##### 2.1.1 Modelo de Dominio

Es una representación visual de conceptos y objetos del mundo real no de los componentes de software, muestra al usuario los conceptos fundamentales que se manejan en el dominio permitiéndole una mejor comprensión de los mismos, además se le considera en RUP un subconjunto del llamado modelo de objetos del negocio. Debido a la poca estructuración de los procesos de negocio se plantea desarrollar un modelo de dominio descrito mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en el sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema. Para realizar este modelo es preciso capturar los tipos más importantes de conceptos en el contexto del sistema y representar las relaciones que existen entre ellos (33).

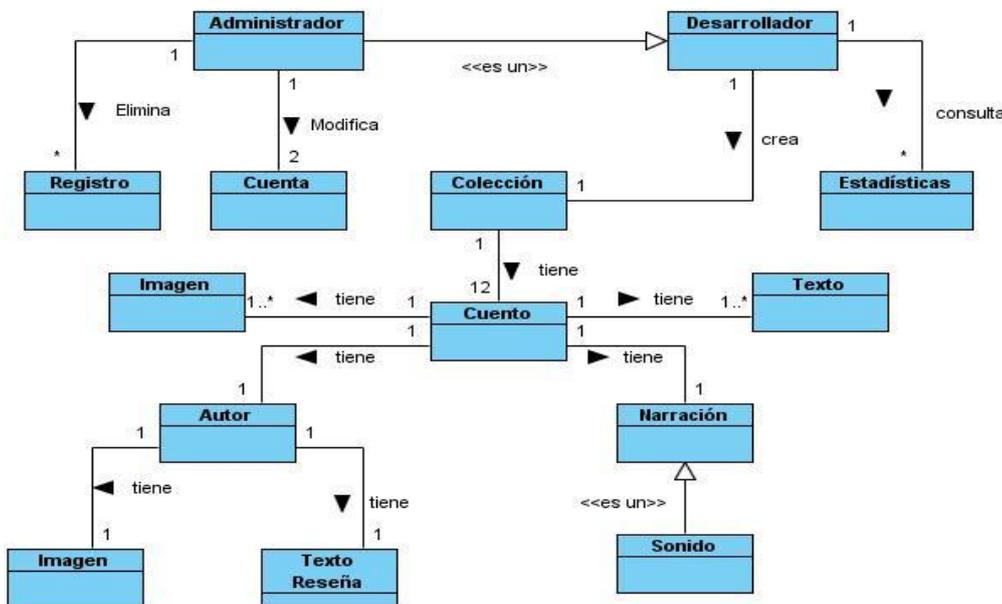


Fig. 8 Modelo de Dominio.

### 2.1.2 Conceptos del Modelo de Dominio.

Con el objetivo de lograr una mejor comprensión de los elementos del modelo de dominio a continuación se muestra una breve explicación de los diferentes conceptos identificados y en qué consiste cada uno de ellos.

- ✓ **Administrador**: Hace referencia a la persona que posee el rol de administrador, este tiene todos los permisos para realizar cualquier cambio dentro de la Herramienta de Administración de Cuentos.
- ✓ **Desarrollador**: Hace referencia a la persona que posee el rol de desarrollador, y tiene permisos para realizar cambios en los cuentos de la colección cargada en la Herramienta de Administración de Cuentos, pero no en las trazas registradas en la herramienta que se propone.
- ✓ **Colección**: Conjunto de 12 cuentos de diferentes autores cada uno de ellos, destinados a niños comprendidos entre 3 y 5 años de edad.
- ✓ **Cuento**: Breve narración de hechos reales o imaginarios protagonizada por un pequeño número de personajes y contiene un argumento sencillo.
- ✓ **Texto**: En esta interfaz se cuenta el texto de los cuentos infantiles que representarán los distintos fragmentos del cuento que se evidenciarán en el producto.
- ✓ **Imagen**: Las imágenes permiten mostrar lo que se está narrando en un cuento seleccionado, y en el caso del autor muestra la imagen asociada al mismo.
- ✓ **Narración**: A través de esta interfaz se escucharán los cuentos narrados por los especialistas encargados de esta tarea.
- ✓ **Sonido**: Es la media que permite escuchar los sonidos de la narración del cuento seleccionado por el usuario.
- ✓ **Autor**: Se especifican los datos más importantes del autor en forma de texto del cuento seleccionado por el usuario.
- ✓ **Texto reseña**: Texto que contiene los datos del autor del cuento seleccionado por el usuario.
- ✓ **Cuenta**: Permite solo a la persona que posea el rol de administrador modificar la contraseña, mediante la cual se tiene acceso al sistema.
- ✓ **Registro**: Muestra todos los cambios realizados en la multimedia “Mis Mejores Cuentos”, estos registros solo podrán ser borrado por el rol administrador.
- ✓ **Estadísticas**: Listado de cuentos con todos sus elementos presentes dentro de la colección existente.

### 2.1.3 Especificación de Requisitos

En el presente trabajo para la captura de los requisitos se emplearon las técnicas de entrevistas, cuestionarios y la tormenta de ideas aplicadas a los desarrolladores y líderes del equipo de trabajo que realizó la multimedia “Mis Mejores Cuentos”.

**Requisitos Funcionales:** Estos requisitos describen la funcionalidad o los servicios que se espera que el sistema cumpla, sus entradas y salidas así como excepciones.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

- RF1 - Autenticar usuario.
- RF2 - Mostrar Registros.
- RF3 - Exportar Registros a PDF.
- RF4 - Eliminar Registros.
- RF5 - Mostrar Estadísticas.
- RF6 - Exportar Estadísticas a PDF.
- RF7 - Mostrar datos del cuento.
  - RF7.1 - Mostrar título del cuento.
  - RF7.2 - Mostrar cantidad de elementos de autor.
  - RF7.3 - Mostrar cantidad de textos.
  - RF7.4 - Mostrar cantidad de narraciones.
  - RF7.5 - Mostrar cantidad de imágenes.
- RF8 - Gestionar Texto reseña.
  - RF8.1 - Modificar texto reseña.
  - RF8.3 - Eliminar texto reseña.
- RF9 – Gestionar imagen autor.
  - RF9.1 - Insertar imagen.
  - RF9.2 - Eliminar imagen.
  - RF 9.3 - Mostrar imagen.
- RF10 – Gestionar texto.
  - RF10.1 - Insertar Texto.
  - RF10.2 - Modificar Texto.
  - RF10.3 - Eliminar Texto.
  - RF10.4 - Mostrar listado de texto.
- RF11 – Gestionar narración.
  - RF11.1 - Insertar narración.
  - RF11.2 - Mostrar listado de narraciones.
  - RF11.3 - Eliminar narración.
- RF12- Reproducir narración.
- RF13 – Gestionar imagen.
  - RF13.1 - Insertar imagen.
  - RF13.2 - Eliminar imagen.
  - RF13.3 - Mostrar imagen.
- RF14 - Modificar cuenta.
  - RF14.1 - Cambiar contraseña de administrador.
  - RF14.2 - Cambiar contraseña de desarrollador.
- RF15 - Crear colección.
- RF16 - Guardar colección.
- RF17 - Cargar colección.

RF 18- Gestionar Título

RF 18.1 - Insertar título.

RF 18.2 - Modificar título.

RF 19 - Modificar portada de cuento.

**Requisitos No Funcionales:** Son los requisitos que definen otras características que debe cumplir el sistema que influyen en la calidad del producto final, y no están relacionados con las funcionalidades del software.

### **Apariencia**

✓ Sistema diseñado para una resolución de 1024x768.

### **Fiabilidad**

Disponibilidad: La información estará disponible para todos los usuarios que accedan al sistema las 24 horas del día y los 7 días de la semana.

### **Requisitos de software**

- ✓ El sistema debe funcionar en PC con sistema operativo Microsoft Windows XP o versiones superiores o Linux.
- ✓ Para el funcionamiento del sistema debe estar instalado como navegador Mozilla Firefox a partir de la versión 3.
- ✓ Para el funcionamiento del sistema en el navegador Mozilla Firefox debe estar instalado para la reproducción de sonido plugins QuickTime.

### **Requisitos de hardware**

- ✓ 512 MB de RAM (recomendado 1GB de RAM o mayor).
- ✓ Procesador Pentium con una frecuencia superior a 1,66 GHz.
- ✓ Lector de CD/DVD (se acepta cualquier marca).

### **Restricciones de diseño e implementación**

- ✓ Se utilizarán como herramientas de modelado Visual Paradigm, y de desarrollo deberán ser empleadas Aptana Studio y Zend Studio.
- ✓ Se utilizarán como lenguajes de desarrollo web Javascript Script, PHP, HTML y CSS.
- ✓ Como framework serán utilizadas las librerías Prototype y JQuery de Javascript.
- ✓ Las imágenes serán en formato .jpg, .gif o .png, de tamaño no mayor de 2MB.
- ✓ Los sonidos serán en formato .wav, de tamaño no mayor de 2MB.
- ✓ Los textos serán tecleados o copiados por el usuario en un cuadro de texto.
- ✓ La Herramienta de Administración de Cuentos no presentará pacotillas.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

## Seguridad

- ✓ La seguridad del sistema se controlará a partir de permisos por nivel de roles, definidos en Administrador o Desarrollador.
- ✓ Para acceder al sistema el usuario debe estar autenticado con un rol determinado.
- ✓ El sistema debe registrar de manera automática las trazas de los usuarios.
- ✓ Las contraseñas de autenticación deben ser fuertes.

### 2.1.4 Actores del Sistema

Actor	Justificación
Administrador	Representa a una persona con máximos privilegios para realizar cualquier cambio en la multimedia “Mis Mejores Cuentos” y en los registros almacenados en la Herramienta de Administración de Cuentos.
Desarrollador	Representa a cualquier persona que tenga el rol de desarrollador y requiera realizar cambios solo en los cuentos de la colección que esté cargada en el sistema sin poder cambiar de colección.

### 2.1.5 Diagrama de Casos de Uso del Sistema (DCUS).

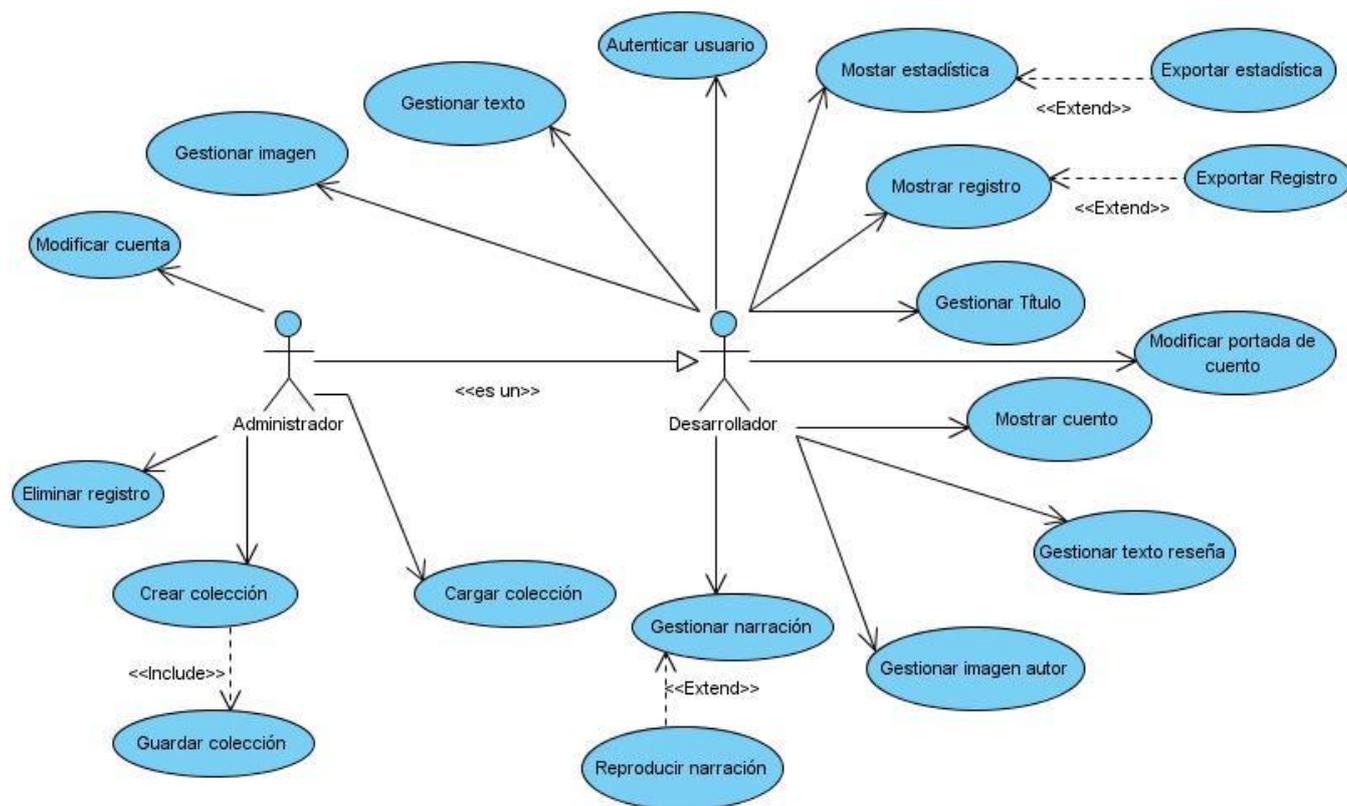


Fig. 9 Diagrama de Casos de Uso del Sistema.

### 2.1.6 Descripción de Casos de Uso del Sistema.

En este epígrafe se muestran las descripciones resumidas de los Casos de Usos (34), para mayor comprensión de los mismos de manera extendida consultar el Anexo 1 donde se encuentra la descripción textual de los más significativos para la Herramienta de Administración de Cuentos.

<b>Caso de Uso:</b>	Autenticar Usuario.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador o el desarrollador introducen su contraseña para que el sistema le permita realizar las funcionalidades definidas para cada rol.
<b>Precondiciones:</b>	Ejecutar la aplicación.
<b>Referencias:</b>	RF1.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Modificar cuenta.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador solicita cambiar su contraseña o la del desarrollador.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador.
<b>Referencias:</b>	RF14.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Crear colección.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador desea crear una nueva colección.

<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador.
<b>Referencias:</b>	RF15.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Cargar colección.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador desea cargar una colección que se encuentre guardada en el sistema.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador y se debe haber guardado alguna colección antes de cargarla.
<b>Referencias:</b>	R17.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Guardar colección.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador desea guardar una colección para luego poder cargarla.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador.
<b>Referencias:</b>	RF16.
<b>Prioridad:</b>	Secundario.
<b>Nivel:</b>	Subfunción.

<b>Caso de Uso:</b>	Eliminar Registro.
<b>Actores:</b>	Administrador.

<b>Resumen:</b>	El caso de uso inicia cuando el administrador desea eliminar todos los registros que contienen las trazas con los cambios realizados en el sistema.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador.
<b>Referencias:</b>	RF4.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Mostrar cuento.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el administrador o el desarrollador desean ver los siguientes datos de un cuento seleccionado: <ul style="list-style-type: none"> <li>- Título del cuento.</li> <li>- Cantidad de elementos de autor.</li> <li>- Cantidad de textos.</li> <li>- Cantidad de narraciones.</li> <li>- Cantidad de imágenes.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado en el sistema.
<b>Referencias:</b>	RF7.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Gestionar texto reseña.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Modificar texto reseña.</li> <li>- Eliminar texto reseña.</li> </ul>

<b>Precondiciones:</b>	El usuario debe haberse autenticado y haber seleccionado el cuento que pertenece al autor que desea gestionar.
<b>Referencias:</b>	RF8.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Gestionar imagen autor.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Insertar imagen autor.</li> <li>- Eliminar imagen autor.</li> <li>- Ver imagen autor.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado y haber seleccionado el cuento que pertenece al autor que desea gestionar.
<b>Referencias:</b>	RF9.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Gestionar narración.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario autenticado desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Insertar narración.</li> <li>- Eliminar narración.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador o desarrollador en la aplicación.
<b>Referencias:</b>	RF11.

<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Reproducir narración.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea escuchar la narración de un cuento seleccionado.
<b>Precondiciones:</b>	El usuario debe haberse autenticado en la aplicación y debe existir una narración para reproducirla.
<b>Referencias:</b>	RF12.
<b>Prioridad:</b>	Secundario.
<b>Nivel:</b>	Subfunción.

<b>Caso de Uso:</b>	Gestionar Texto.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario autenticado desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Insertar texto.</li> <li>- Modificar texto.</li> <li>- Eliminar texto.</li> <li>- Consultar Listado de texto.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado en la aplicación.
<b>Referencias:</b>	RF10.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Gestionar imagen.
---------------------	-------------------

<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario autenticado desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Insertar imagen.</li> <li>- Eliminar imagen.</li> <li>- Mostrar imagen.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado en la aplicación.
<b>Referencias:</b>	RF13.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Gestionar título.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario autenticado desea realizar alguna de las siguientes acciones dentro del sistema: <ul style="list-style-type: none"> <li>- Insertar título.</li> <li>- Modificar título.</li> </ul>
<b>Precondiciones:</b>	El usuario debe haberse autenticado en la aplicación.
<b>Referencias:</b>	RF18.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Modificar portada de cuento.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario autenticado desea modificar la portada de un cuento seleccionado.
<b>Precondiciones:</b>	El usuario debe haberse autenticado en la aplicación.

<b>Referencias:</b>	RF19.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Mostrar estadística.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea revisar la cantidad de elementos que existen en un cuento de la colección existente en el sistema.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador o desarrollador.
<b>Referencias:</b>	RF5.
<b>Prioridad:</b>	Secundario.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Exportar estadística.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea guardar las estadísticas de una colección como PDF.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador o desarrollador.
<b>Referencias:</b>	RF6.
<b>Prioridad:</b>	Secundario.
<b>Nivel:</b>	Subfunción.

<b>Caso de Uso:</b>	Mostrar registro.
---------------------	-------------------

<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea revisar los últimos cambios que se han realizado en el sistema.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador o desarrollador.
<b>Referencias:</b>	RF2.
<b>Prioridad:</b>	Crítico.
<b>Nivel:</b>	Usuario.

<b>Caso de Uso:</b>	Exportar registro.
<b>Actores:</b>	Administrador, Desarrollador.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario desea guardar los registros de cambios en formato PDF.
<b>Precondiciones:</b>	El usuario debe haberse autenticado como administrador o desarrollador.
<b>Referencias:</b>	RF3.
<b>Prioridad:</b>	Secundario.
<b>Nivel:</b>	Subfunción.

## 2.2 Análisis y Diseño

Cada clase del diseño representa una abstracción con una correspondencia en la implementación. Este modelo brinda una completa idea de lo que es el software y constituye un punto de partida para las actividades de implementación.

### 2.2.1 Patrón arquitectónico.

#### 2.2.1.1 Modelo Vista Controlador (MVC).

El patrón MVC separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres partes diferentes (35).

**Modelo:** El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requisitos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

**Vista:** Maneja la visualización de la información.

**Controlador:** Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual (36).

Dado que la vista se encuentra separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente a esta ventaja se le denomina soporte de vistas múltiples por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos, mostrado de maneras diferentes. Además de esta ventaja se encuentra la llamada adaptación al cambio en la que los requisitos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo. Este patrón sentó las bases para especializaciones ulteriores, tales como Page Controller y Front Controller (36). MVC aporta una construcción de software mantenible, en la que se pueden localizar de forma ágil los errores, ofreciendo maneras sencillas para probar el correcto funcionamiento del sistema además que facilita el mantenimiento en caso de errores y el escalamiento de la aplicación en caso de ser requerido. Supone un diseño modular, y muy poco acoplado, favoreciendo la reutilización.

## **2.2.2 Patrones de diseño.**

### **2.2.2.1 Patrones GRASP.**

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen la base del cómo se diseñará el sistema y se aplican en los primeros momentos del diseño.

#### **✓ Patrón Experto**

Permite asignar una responsabilidad al experto en información. Nos indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantendrá encapsulada. Entre sus ventajas están:

- Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.

- El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas, las cuales son fáciles de comprender y mantener.

#### ✓ Patrón Creador

Define quién debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Dentro de sus ventajas están:

- Guiar la asignación de responsabilidades relacionadas con la creación de objetos.
- Se encuentra un creador que necesite conectarse al objeto creado en alguna situación, eligiéndolo como el creador, se favorece el bajo acoplamiento.

#### ✓ Patrón Bajo Acoplamiento

Este patrón define como dar soporte a una dependencia escasa y a un aumento de la reutilización, enfocándose en asignar una responsabilidad para mantener bajo acoplamiento. El bajo acoplamiento estimula la asignación de responsabilidades de forma tal que, la inclusión de estas no incremente el acoplamiento, creando clases más independientes y con mayor resistencia al impacto de los cambios, que aumentan la productividad y la posibilidad de reutilización. Es válido aclarar que este patrón no puede verse de forma independiente a los patrones Experto y Alta Cohesión, sino más bien incluirse como otro de los principios del diseño que influyen de forma determinante a la hora de la asignación de responsabilidades.

#### Ventajas

- No afectan los cambios en otros componentes.
- Fácil de entender de manera aislada.
- Conveniente para reutilizar.

#### ✓ Patrón Alta Cohesión

El propósito de este patrón es asignar una responsabilidad de manera que la cohesión permanezca alta. Posibilitando mucha facilidad de mantenimiento, comprensión y uso. Brinda un alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos de las clases que componen el software.

#### Ventajas

- Se incrementa la claridad y facilita la comprensión del diseño.
- Se simplifican el mantenimiento y las mejoras.
- Se soporta a menudo bajo acoplamiento.

### ✓ Patrón Controlador

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el que recibe los datos del usuario y el que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Además de estos patrones se utilizan otros más específicos acordes a la aplicación que se desea desarrollar, como son:

#### 2.2.2.2 Patrones GOF

Gang-of-Four es el nombre con el que se conoce comúnmente a los autores del libro Design Patterns. Un patrón GoF es una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular.

### ✓ Patrón Singleton.

El Singleton es quizás el más sencillo de los patrones que se presentan en el catálogo de los patrones GoF. Es también uno de los patrones más conocidos y utilizados. Su propósito es asegurar que sólo exista una instancia de la clase. Cuando varios elementos distintos precisan referenciar a un mismo elemento y se desea asegurar que no hay más de una instancia de ese elemento, simplemente el patrón Singleton determina un punto de acceso global a esta clase o instancia garantizando resolver el problema.

### ✓ Patrón Facade o Fachada.

Su objetivo es proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. Lo usamos cuando se pretende estructurar en capas el subsistema, para que cada capa tenga su propia fachada. Lo que nos permitirá usar las clases del subsistema que necesitemos sin ningún obstáculo. De esta forma se puede elegir entre facilidad de uso y generalidad.

### ✓ Patrón Observer.

Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos. En la aplicación serán utilizados para actualizar determinados elementos que dependen directamente de los datos de otros, específicamente para actualizar la interfaz de usuario ante un cambio en los datos cargados desde los ficheros Json.

### 2.2.3 Clases del Diseño.

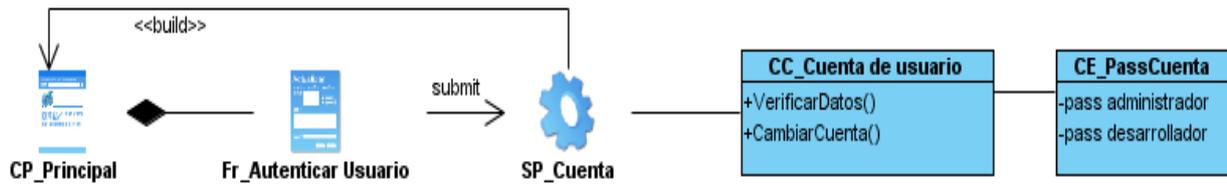


Fig. 10 DCD\_CU Autenticar usuario.



Fig. 11 DCD\_CU Modificar cuenta.

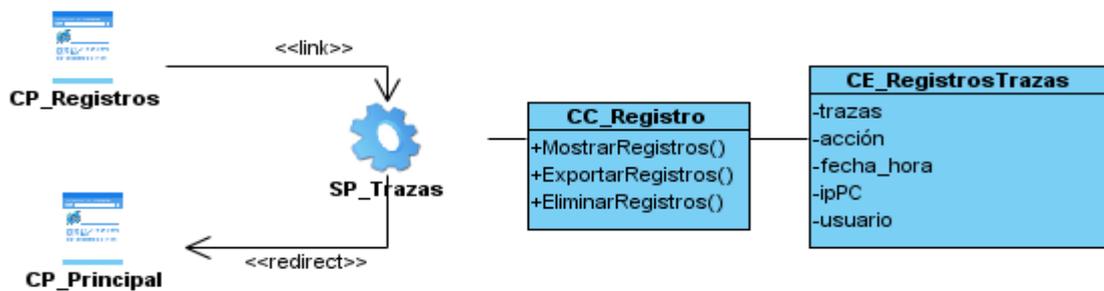


Fig. 12 DCD\_CU Eliminar registro.

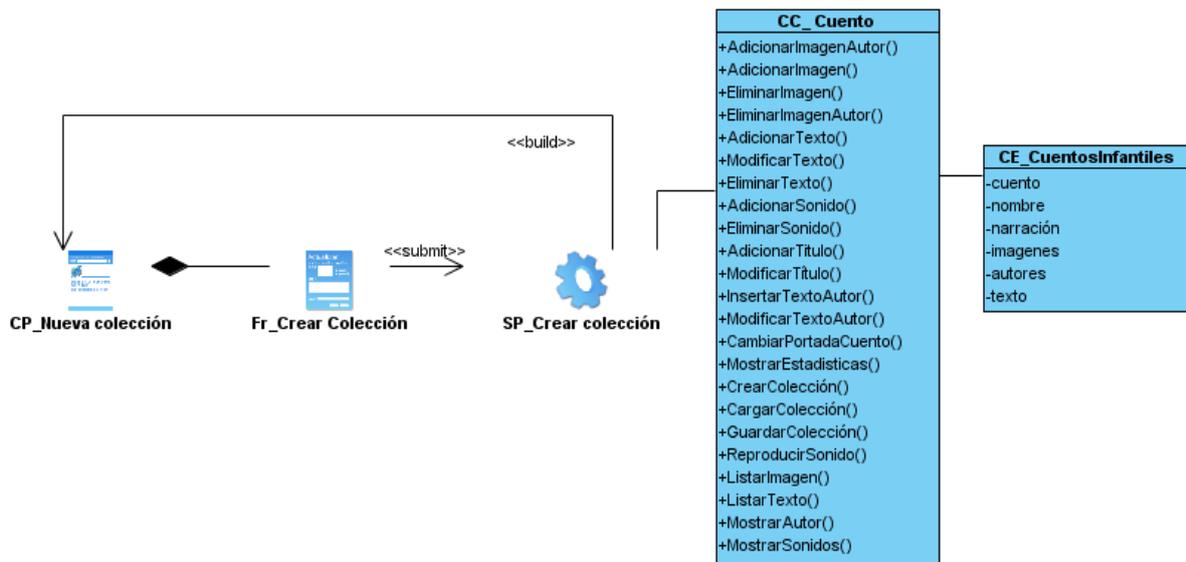


Fig. 13 DCD\_CU Crear colección.

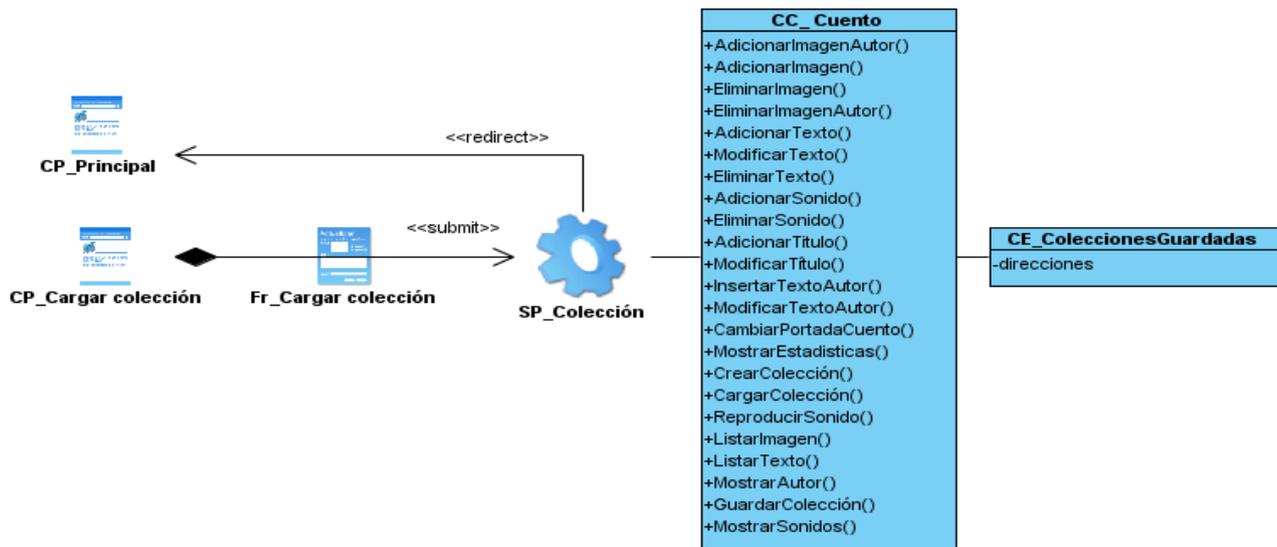


Fig. 14 DCD\_CU Cargar colección.

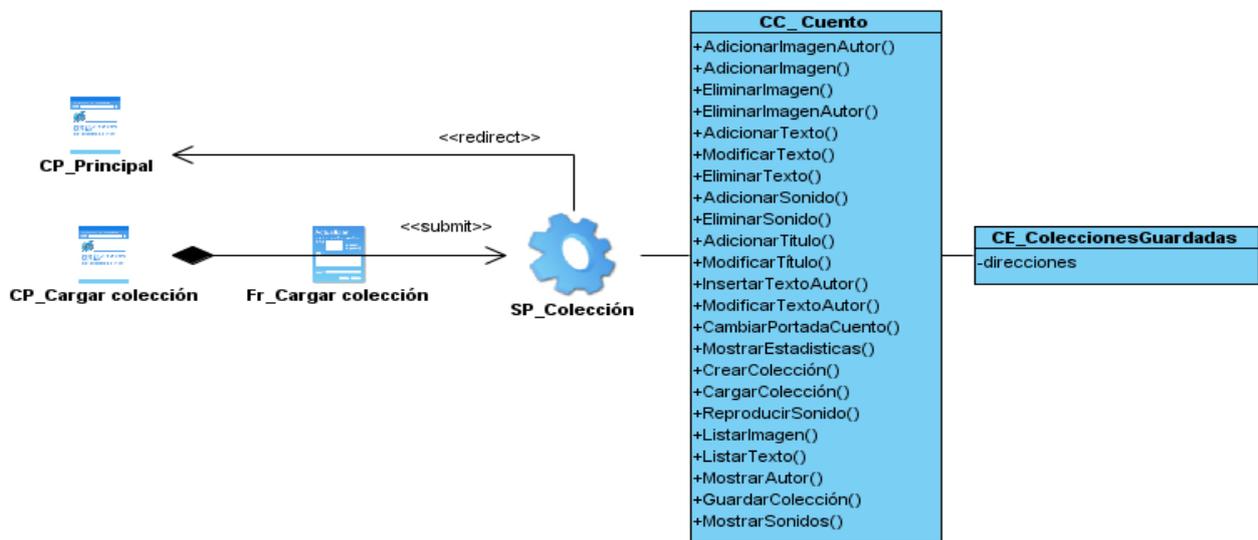


Fig. 15 DCD\_CU Guardar colección.

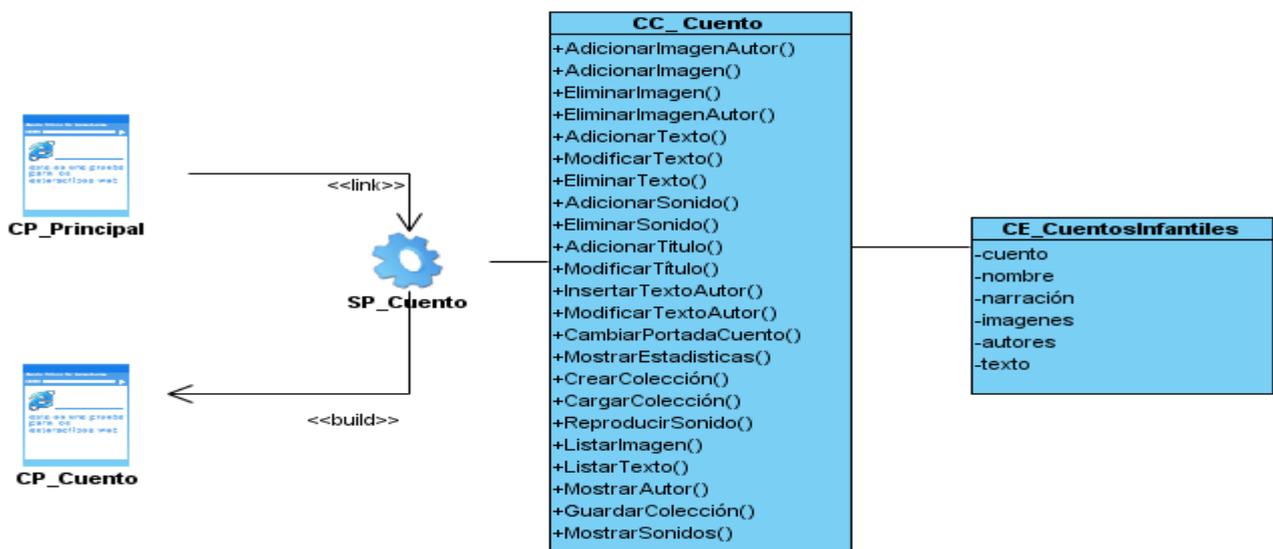


Fig. 16 DCD\_CU Mostrar cuento.

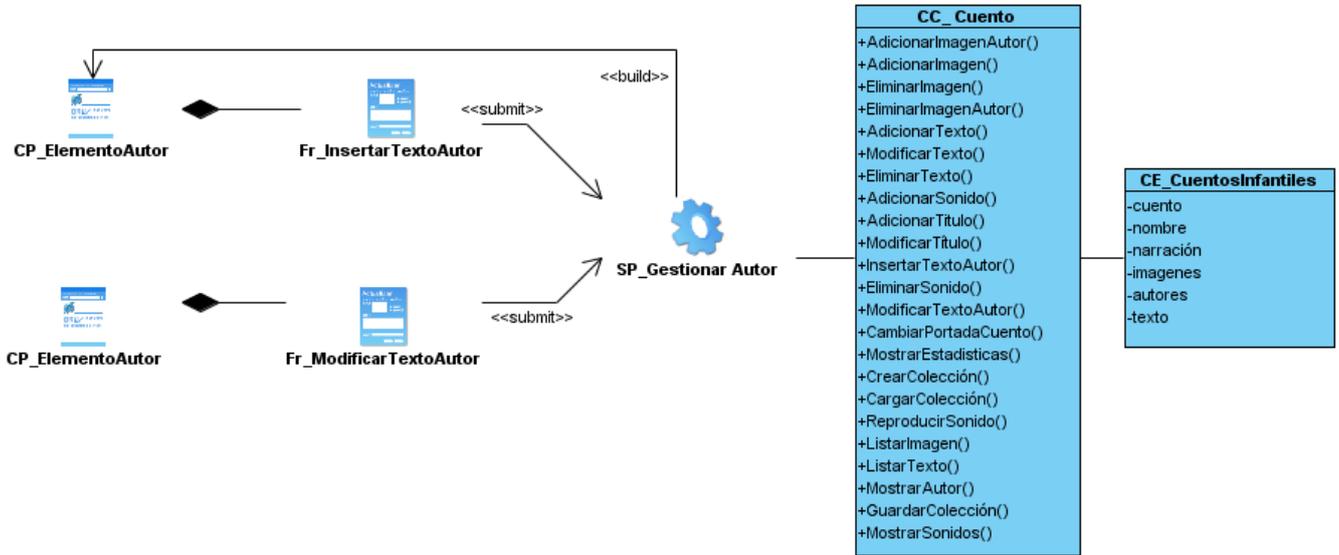


Fig. 17 DCD\_CU Gestionar texto reseña.

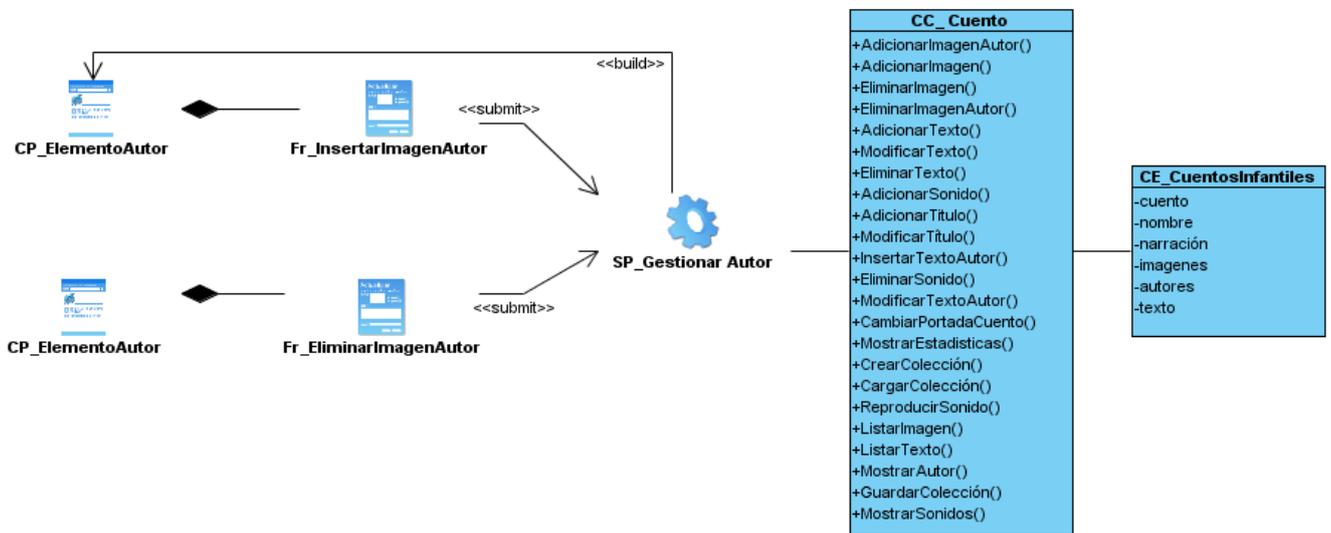


Fig. 18 DCD\_CU Gestionar imagen autor.

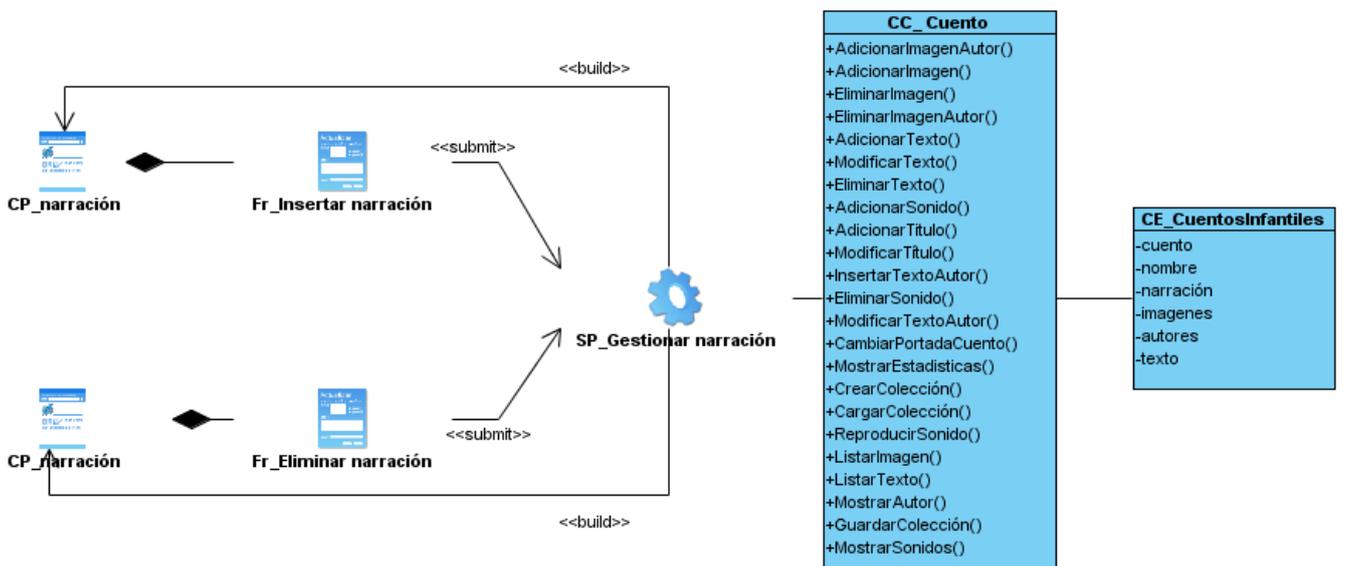


Fig. 19 DCD\_CU Gestionar narración.

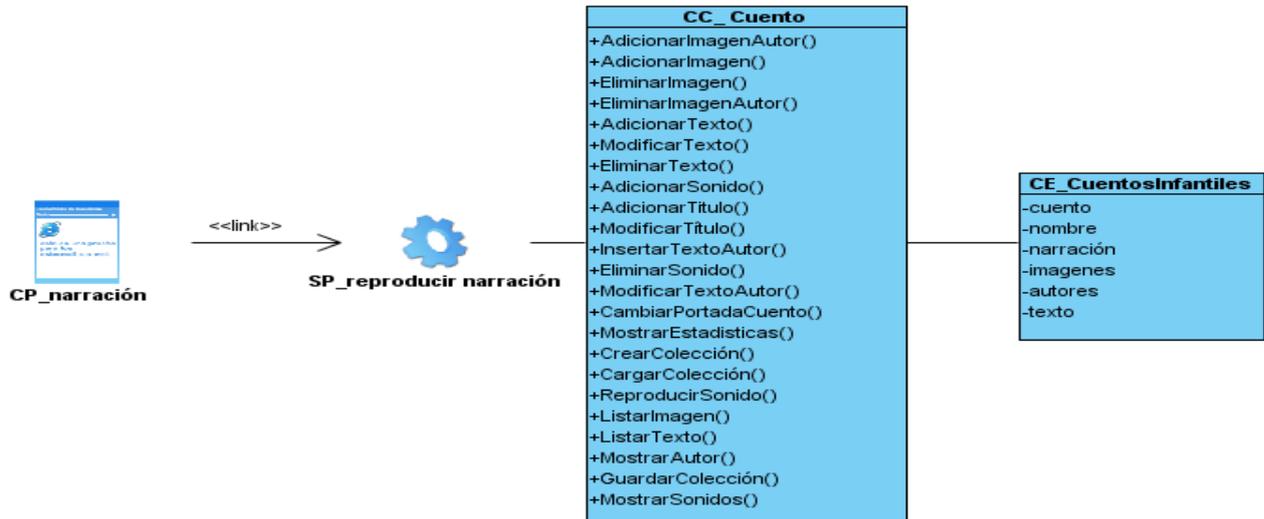


Fig. 20 DCD\_CU Reproducir narración.

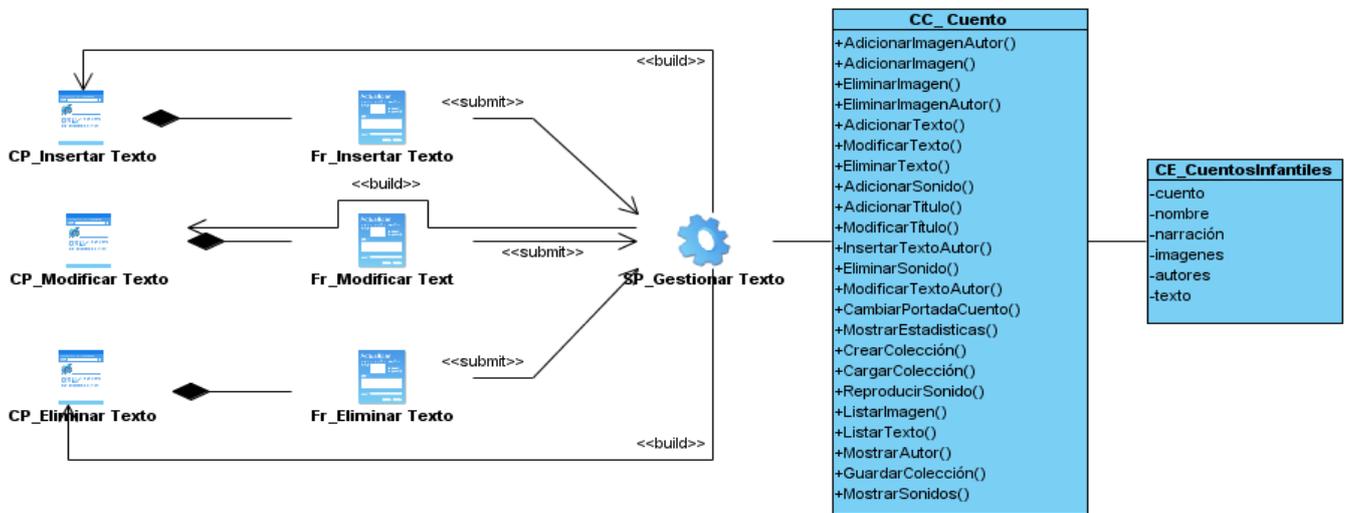


Fig. 21 Gestionar texto.

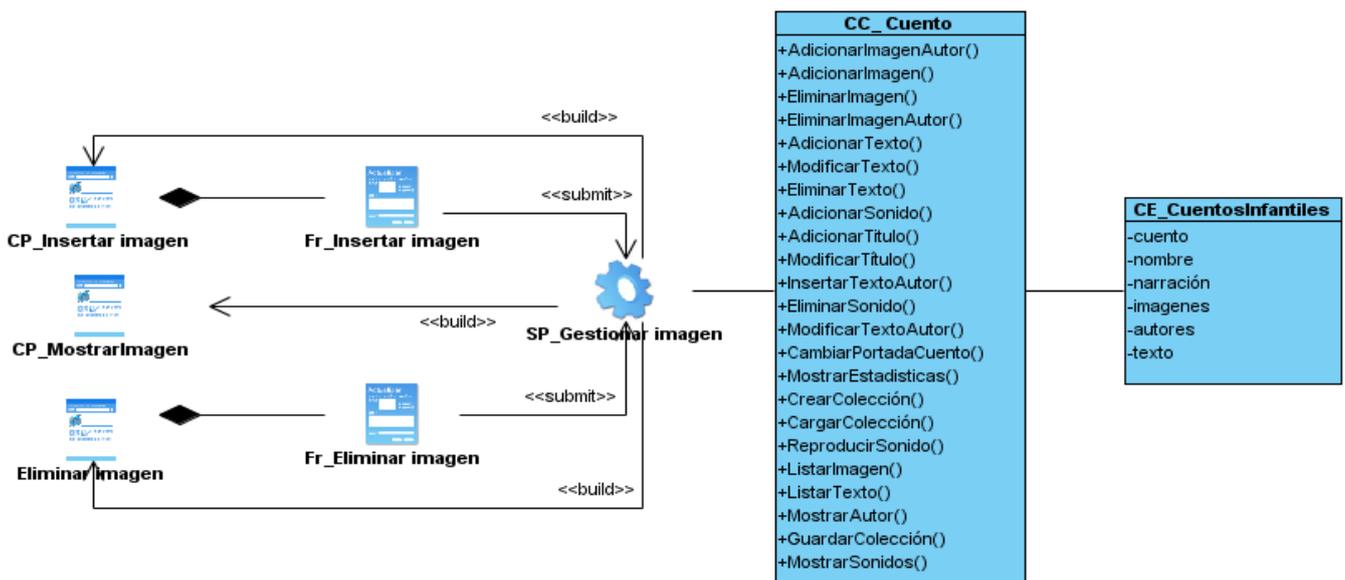


Fig. 22 DCD\_CU Gestionar Imagen.

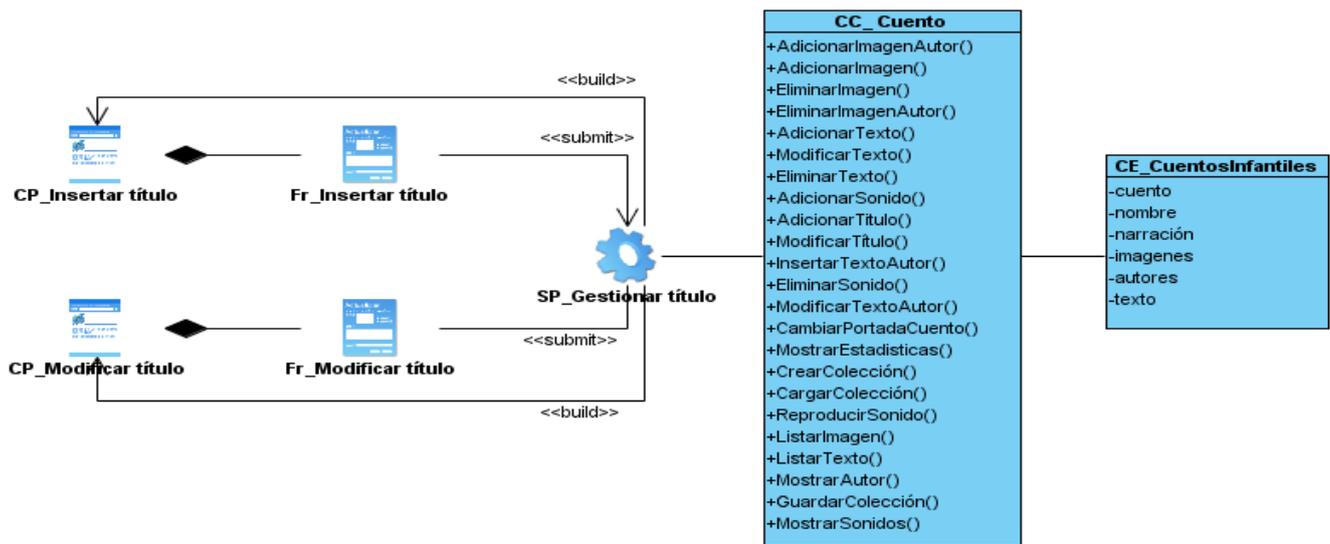


Fig. 23 Gestionar título.

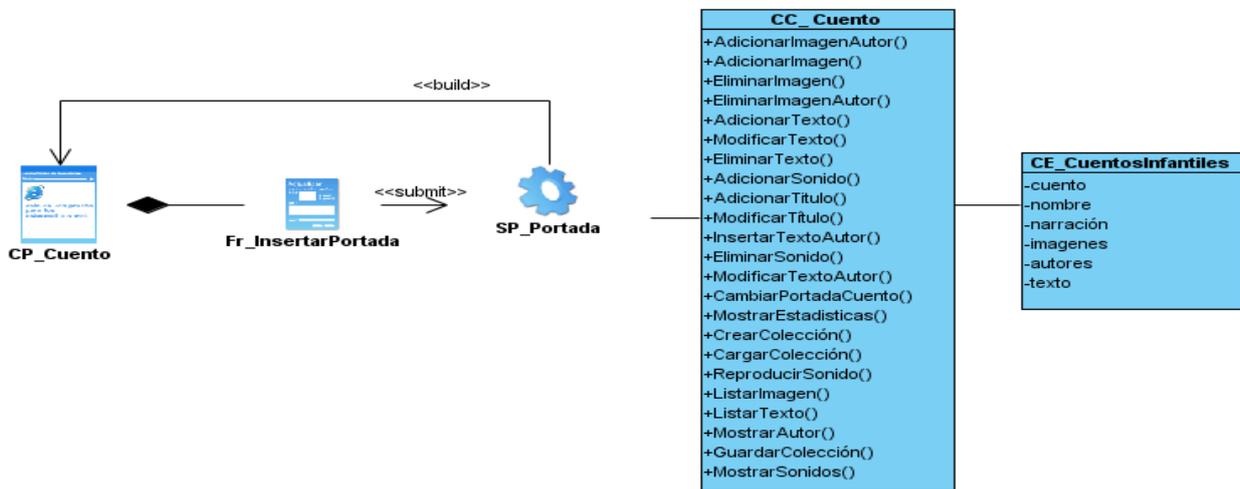


Fig. 24 DCD\_CU Modificar portada cuento.

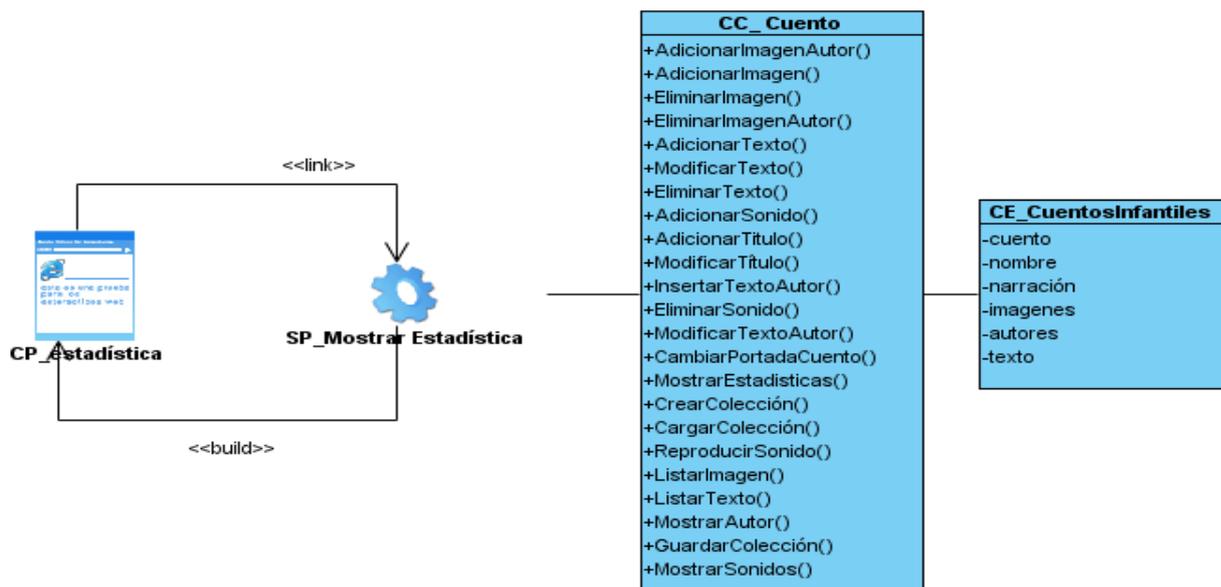


Fig. 25 DCD\_CU Mostrar estadística.

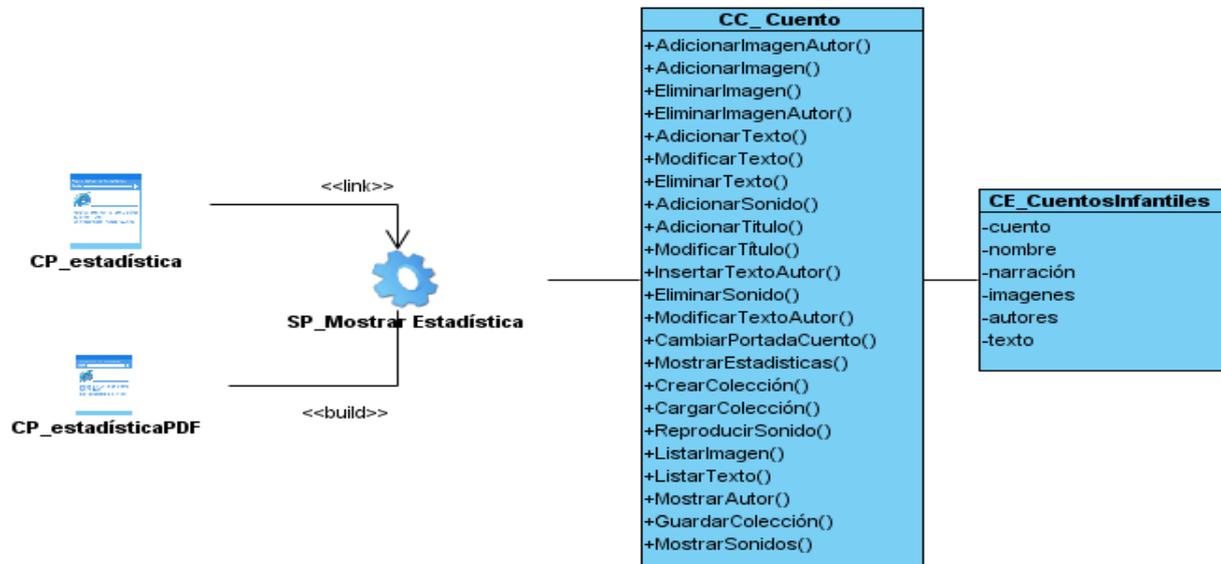


Fig. 26 DCD\_CU Exportar estadística PDF.

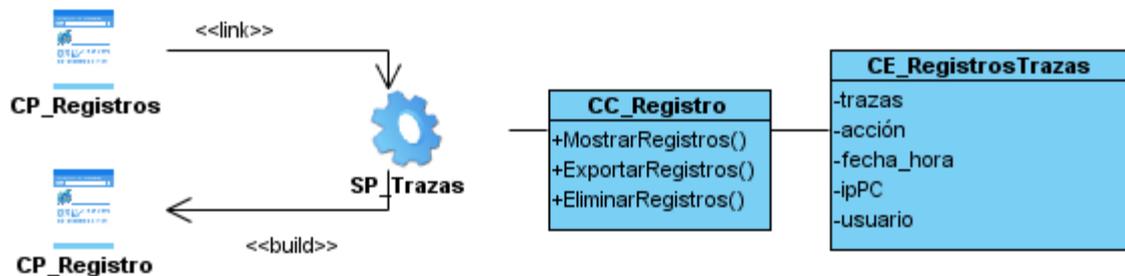


Fig. 27 DCD\_CU Mostrar registro.

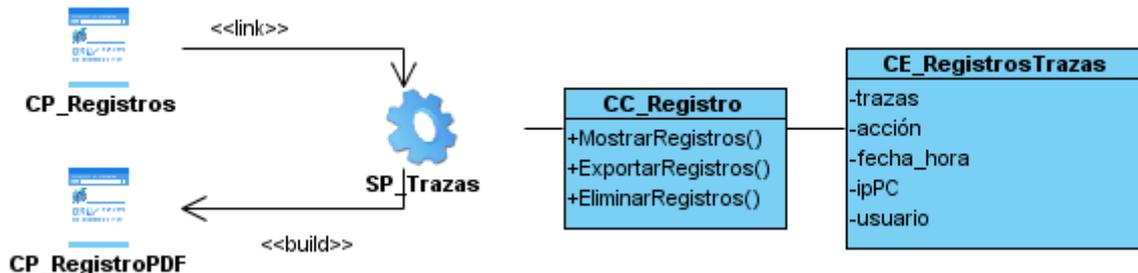


Fig. 28 DCD\_CU Exportar registro PDF.

## 2.2.4 Modelo de Clases Persistentes.

El sistema no trabaja con una BD física pero si cuenta con un archivo de intercambio de datos .json en el cual se almacenan los datos que mostraremos a continuación y que son los considerados persistentes. La capacidad que posee un objeto de mantener su valor en el espacio y en el tiempo es lo que se conoce como persistencia.

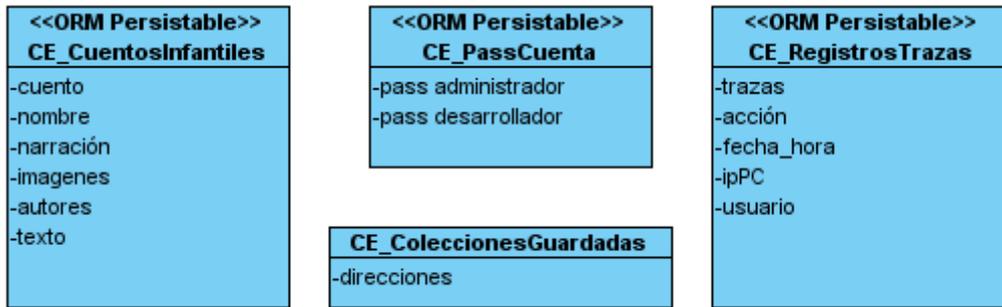


Fig. 29 Diagrama de Clases Persistentes.

## 2.3 Implementación.

La implementación comienza con los resultados del diseño, se describen los componentes a construir, así como la organización y dependencia entre nodos físicos en los que funcionará la aplicación, a través de los Diagramas de Componentes y de Despliegue.

### 2.3.1 Diagrama de componentes.

Los Diagramas de Componentes muestran los componentes de software (ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares) y las relaciones que existen entre ellos.

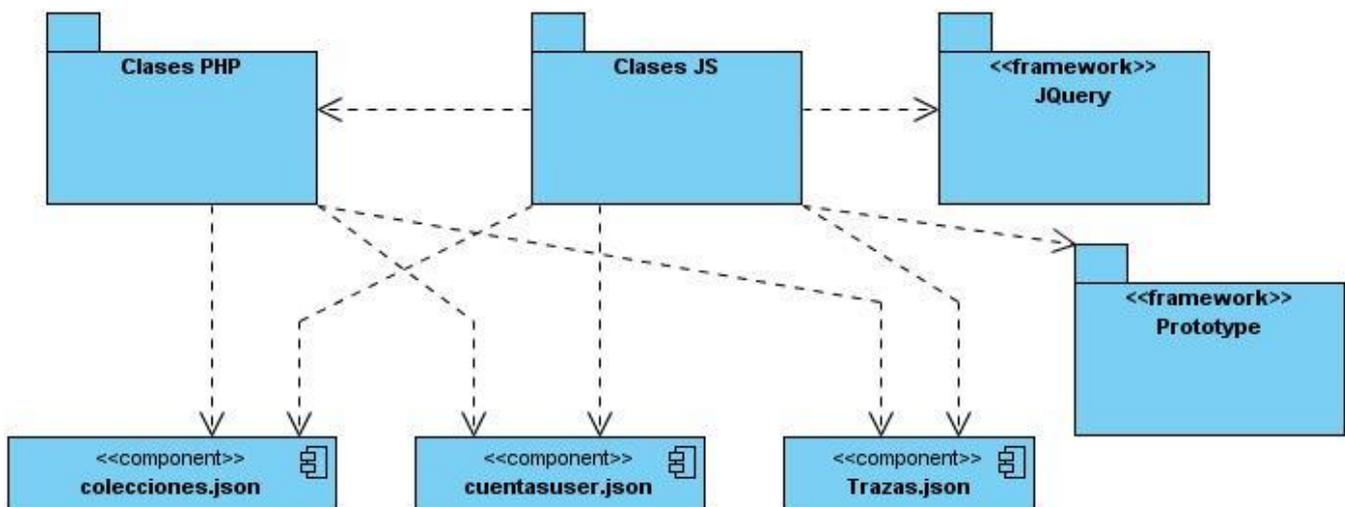


Fig. 30 Diagrama de componentes general.

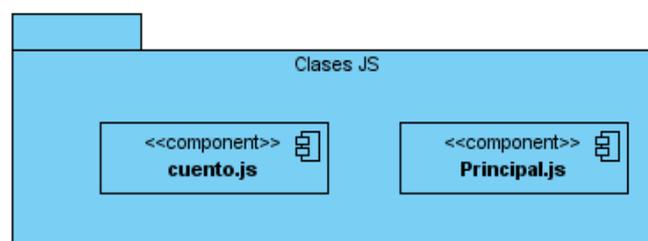


Fig. 31 Diagrama de componentes del paquete clases JS.

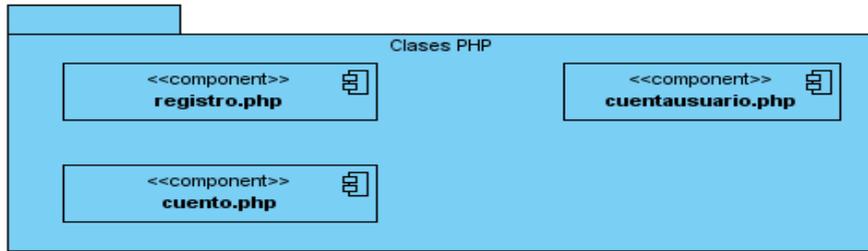


Fig. 32 Diagrama de componentes del paquete clases PHP.

### 2.3.2 Diagrama de Despliegue.

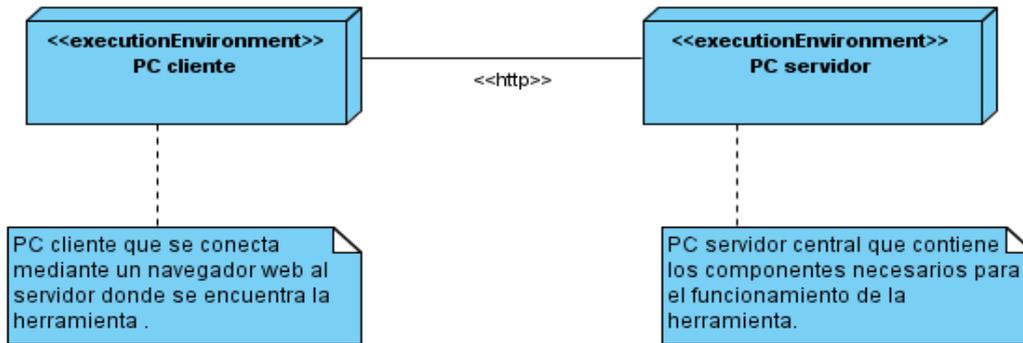


Fig. 33 Diagrama de despliegue.

### 2.3.3 Interfaz de la aplicación.



Fig. 34 Interfaz Página principal.



Fig. 35 Interfaz Inicial de una colección.



Fig. 36 Interfaz Portada del cuento.



Fig. 37 Interfaz Título del cuento.

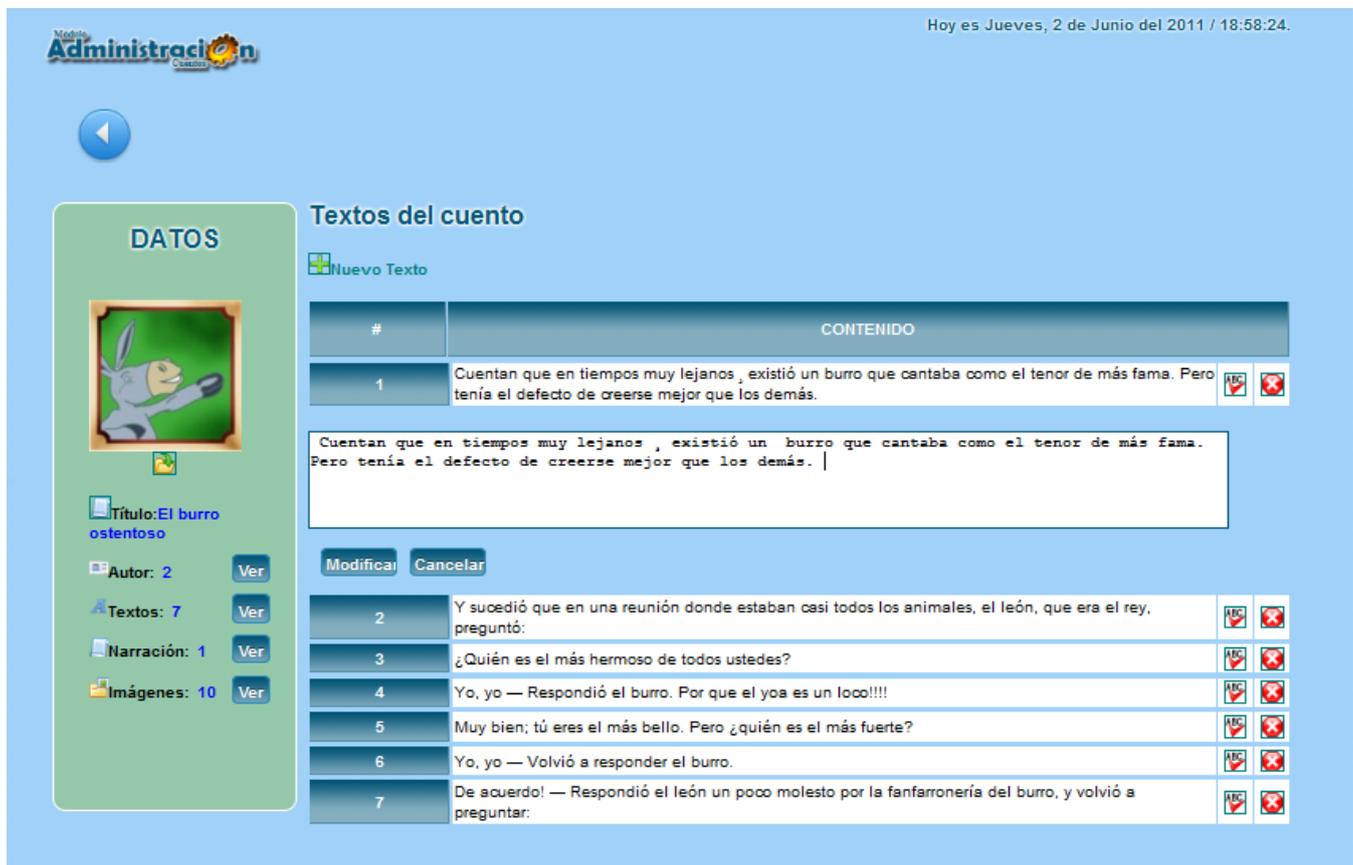


Fig. 38 Interfaz Modificar Texto.

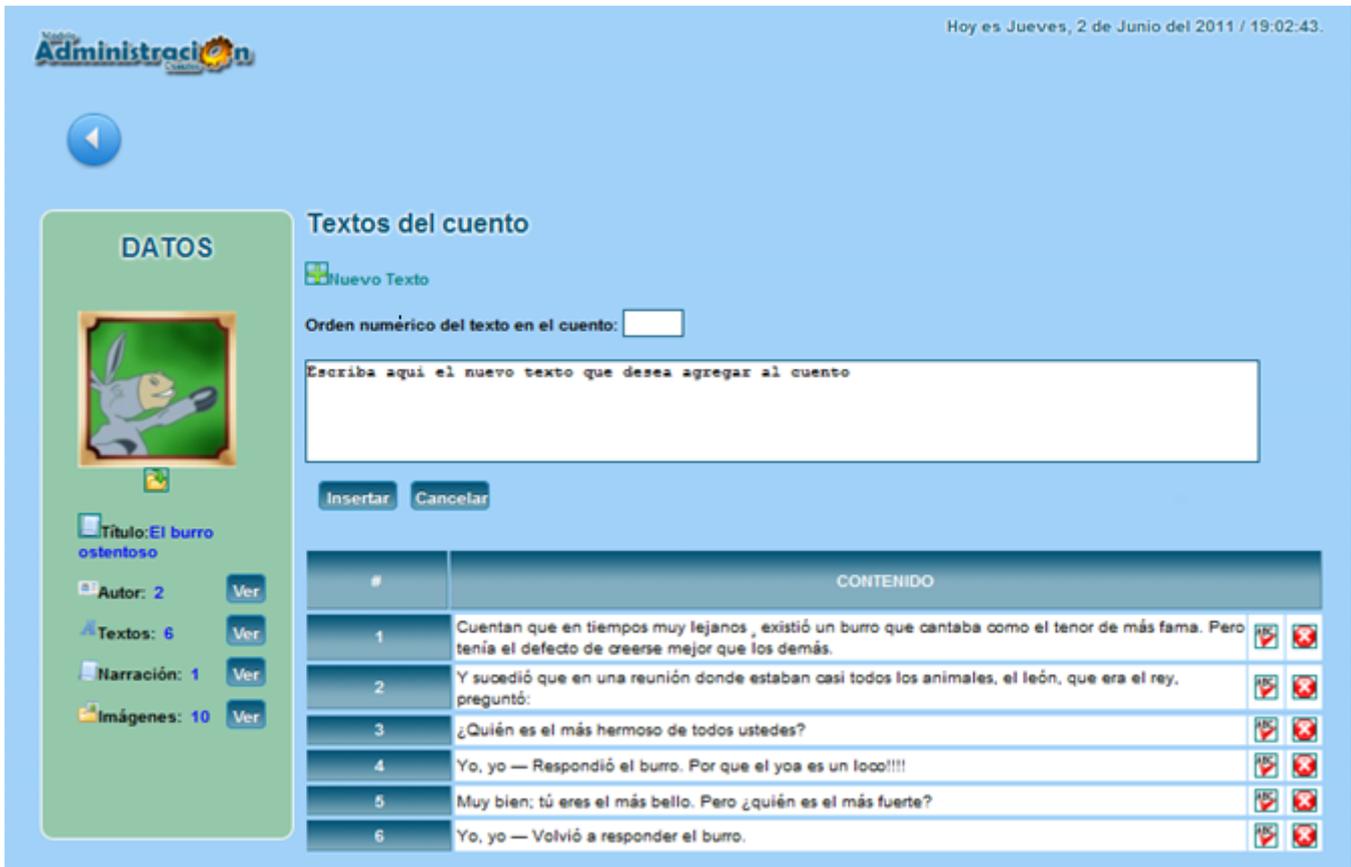


Fig. 39 Interfaz Insertar Texto.

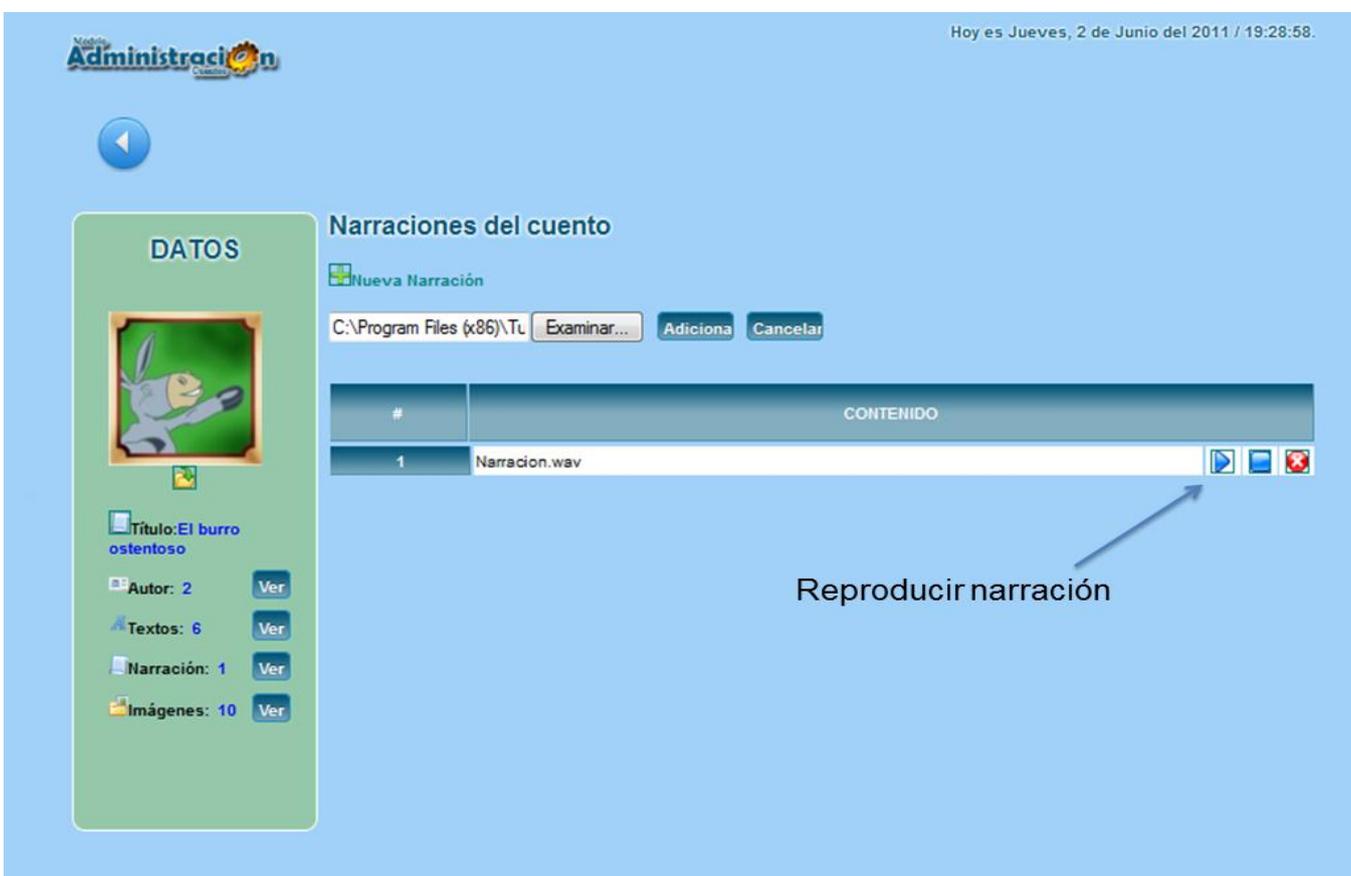


Fig. 40 Interfaz narraciones.



Fig. 41 Interfaz Imágenes.



Fig. 42 Interfaz Cuenta de usuario.

Administrador Hoy es Jueves, 26 de Mayo del 2011 / 11:18:46

Modulo **Administración** Cuentos

Cuenta

Registros

Estadísticas

Nueva Colección

Cargar Colección

Salir

**REGISTROS DE CAMBIOS REALIZADOS POR EL SISTEMA** ✕

Eliminar trazas ← → Exportar Trazas

#	Acciones	IP/PC	Usuario	FechaHora
1	La imagen de autor del cuento <b>El burro ostentoso</b> fue eliminada	127.0.0.1	Administrador	26/05/2011 11:18:31
2	El texto de autor del cuento <b>El burro ostentoso</b> fue modificado	127.0.0.1	Administrador	26/05/2011 11:18:29
3	La imagen número 7 del cuento <b>El burro ostentoso</b> fue eliminada	127.0.0.1	Administrador	26/05/2011 11:18:18
4	El texto número 5 del cuento <b>El burro ostentoso</b> fue eliminado	127.0.0.1	Administrador	26/05/2011 11:18:13
5	Imagen adicionada en la posición 1 del cuento 3	127.0.0.1	Desarrollador	26/05/2011 11:17:25
6	Imagen de autor agregada en el cuento 9	127.0.0.1	Administrador	26/05/2011 11:16:26
7	El nombre del cuento <b>El cocodrilo</b> fue cambiado	127.0.0.1	Administrador	26/05/2011 11:16:04
8	Trazas eliminadas	127.0.0.1	Administrador	26/05/2011 11:15:34

Fig. 43 Interfaz Registros del sistema.

Modulo **Administración** Cuentos Hoy es Jueves, 2 de Junio del 2011 / 19:50:21

Cuenta

Registros

Estadísticas

Nueva Colección

Cargar Colección

Salir

Mis Mejores Cuentos

Fig. 44 Interfaz Nueva colección.



- Cuenta
- Registros
- Estadísticas
- Nueva Colección
- Cargar Colección**
- Salir

Cargar colección		
#	COLECCIONES GUARDADAS	
1	../DATOS/Recuperacion/UltimaVercionJson.json	Cargar
2	../DATOS/Recuperacion/PruebaMovidaArchivos.json	Cargar
3	../DATOS/Recuperacion/coleccion1.json	Cargar
4	../DATOS/Recuperacion/coleccion1.json	Cargar
5	../DATOS/Recuperacion/asd.json	Cargar
6	../DATOS/Recuperacion/M.json	Cargar

Fig. 45 Interfaz Cargar colección.

## 2.4 Validación de la solución propuesta.

El proceso de pruebas es fundamental para detectar errores o fallas en un sistema. Para la herramienta de administración del producto “Mis Mejores Cuentos” se miden los indicadores de estabilidad, eficiencia, escalabilidad y seguridad, que permiten obtener un producto con calidad.

### 2.4.1 Métodos de Prueba.

Para la aplicación de los métodos de prueba es necesario diseñar pruebas en aquellas partes del software donde la probabilidad de fallos sea mayor, para así poder detectar la mayor cantidad de errores que puedan existir, ya que estas se hacen con el propósito de descubrir fallos en el sistema y no para demostrar que el software funciona correctamente.

#### ✓ Pruebas de caja blanca

Se basa en los detalles procedimentales. Se comprueban los caminos lógicos proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado (37).

#### ✓ Pruebas de caja negra.

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software (37).

## 2.4.2 Pruebas aplicadas al software

En el presente trabajo se decide realizar las pruebas de aceptación y de funcionalidad, las cuales pertenecen al método de caja negra y están regidas por los casos de prueba que fueron generados a partir de la especificación de casos de uso. Los casos de prueba presentan los requisitos que se espera cumpla la aplicación, por lo que al escribirlos también se documenta el sistema. Las descripciones de los Casos de Pruebas significativos para el sistema se encuentran de manera íntegra en el Anexo 2.

Se realizaron las siguientes pruebas:

- ✓ Pruebas de aceptación.
- ✓ Pruebas de funcionalidad.

Se realizó una primera iteración en la que se encontraron 4 no conformidades, que representa un 20 % de fallos en las funcionalidades probadas en la aplicación:

- ✓ **NC1. Envío de campos vacíos:** Al enviar los campos de autenticación de usuario vacíos el sistema debe mostrar un mensaje indicando que estos deben ser llenados, lo que no ocurrió en el momento de ser probado el sistema en su primera iteración.
- ✓ **NC2. Envío de campos vacíos:** Al crear una nueva colección al enviar el campo nombre colección vacío el sistema debe mostrar un mensaje indicando llenar el campo, lo que no ocurrió en el momento de ser probado el sistema en su primera iteración.
- ✓ **NC3. Registro de trazas:** A realizar operaciones sobre el sistema se debe guardar la dirección de IP física de donde se están realizando las acciones sobre los cuentos, esta actividad no ocurrió en el momento de ser probado el sistema en su primera iteración.
- ✓ **NC4. Usabilidad:** Al probar las funcionalidades se detectaron faltas de ortografía en 6 de ellas.

Luego fue realizada una segunda iteración en la que no se obtuvieron no conformidades.

## 2.5 Conclusiones del capítulo.

En el presente capítulo se realizó un análisis acerca de las características del sistema, permitiendo identificar requisitos funcionales y no funcionales, y a partir de estos generar el diagrama de casos de uso del sistema, así como la descripción de casos de uso y la definición del modelo de dominio. A partir de estos artefactos se obtuvieron también los modelos de análisis y diseño y los diagramas de clases correspondientes a cada uno, indicando los patrones tanto arquitectónico como de diseño empleados. Se obtuvieron los diagramas de componentes y despliegue proporcionando la descripción de los componentes desarrollados y su organización, además de las interfaces de la herramienta, diseñadas con el propósito de satisfacer las expectativas del usuario y para ello fueron realizadas pruebas de caja blanca y caja negra al sistema obteniendo resultados satisfactorios.

## Conclusiones Generales

Para dar cumplimiento al objetivo general de este trabajo y a la problemática planteada, se realizaron con éxito las tareas trazadas en el comienzo del mismo.

- ✓ La selección de la metodología RUP como guía del proceso de desarrollo fue correcta, pues se documentó todo el ciclo de desarrollo de la herramienta, característica fundamental para un equipo de desarrollo que se encuentra en constante cambio.
- ✓ Se obtuvo una Herramienta de Administración de Cuentos sobre una plataforma web que permite gestionar las medias del Software Educativo “Mis Mejores Cuentos”, sin necesidad de acceder al código fuente del producto.
- ✓ Se validó la herramienta de administración del producto “Mis Mejores Cuentos”, a partir de los indicadores de estabilidad, eficiencia, escalabilidad, usabilidad y seguridad, donde se demostró que la herramienta cumple con los requisitos propuestos.

## Recomendaciones

A partir de los resultados obtenidos en la realización de este trabajo de diploma se proponen las siguientes recomendaciones.

- ✓ Consultar y/o utilizar la documentación generada en el presente trabajo como base en la realización de proyectos con características similares.
- ✓ Una vez implementado el módulo de entretenimiento de la multimedia Web “Mis Mejores Cuentos” integrarlo a la herramienta de administración con el objetivo de gestionar las medias de dicho módulo.
- ✓ Revisar la arquitectura de información de la estructura de las colecciones que posibilite mejorar las modificaciones a las mismas dentro de la multimedia Web “Mis Mejores Cuentos”, así como la estructura del fichero “cuentos.json” con el objetivo de establecer todos los componentes dinámicos para modificarlos con la Herramienta de administración.

## Referencias Bibliográficas

1. **Varios, Autores.** Formación de técnicos e investigadores en tecnologías de la información. [aut. libro] Fundesco. Madrid : s.n., 1986.
2. La producción en la UCI. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.uci.cu/?q=node/46>.
3. **Fabelo.** Fabelo. 2008.
4. **Marques, Perez.** 2005.
5. **Cuerda, Xavier García.** Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. [En línea] [Citado el: 10 de Diciembre de 2010.] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-conte..>
6. Ventajas y Desventajas de los CMS. [En línea] Diciembre de 2010. [Citado el: 14 de Enero de 2011.] <http://www.soluciones-web.com.ve/servicios/cms.html..>
7. **Béjar, Juan.** *Sistemas de gestión de contenidos (CMS) para sitios educativos.* 2010.
8. *Gestión de contenidos en el e-learning: acceso y uso de objetos de información como recurso estratégico.* **Duart, Pablo Lara Navarra y Josep Maria.** Montoliu : Revista de Universidad y Sociedad del Conocimiento, Vol. II.
9. **Bianchini.** Definiciones para hipertexto.
10. **Karenia.** Software y Estrategias de Aprendizaje. [En línea] Marzo de 2010. [Citado el: 7 de Enero de 2011.] [http://blogs.rimed.cu/futuro/ ..](http://blogs.rimed.cu/futuro/)
11. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión.* **Piattini.** Rama, Madrid. : s.n.
12. **Calero.** Una explicación de la programación extrema (XP). [En línea] [Citado el: 12 de enero de 2011.] <http://www.apolosoftware.com..>
13. **Billy, Carlos Reynoso.** *Métodos Ágiles en Desarrollo de Software.* Universidad de Buenos Aires : s.n.
14. **Moraga., Prof. Hernán.** *Programación ágil: SCRUM y XP.* . Universidad Mayor de Santiago de Chile. : s.n.
15. **Jiménez, S. V.** *Propuesta de una estructura organizacional del proceso de producción de un software multimedia a través de RUP y una extensión de UML para hipermedia.* Universidad de las Ciencias Informáticas, Ciudad Habana. Cuba. : s.n., 2006.
16. Metodología XP Vs. Metodología RUP. [En línea] [Citado el: 12 de enero de 2011.] [http://metodologiapxvsmetodologiarup.blogspot.com/.](http://metodologiapxvsmetodologiarup.blogspot.com/)
17. **Quirón.** Introducción a UML 2.0. 2005.
18. **autores., Dania Souchay Fabrega y otros.** *Multimedia aprendarte, centralización de información.* La Habana, Cuba. : s.n.
19. Sitio oficial de PostgreSQL. [En línea] 2010. <http://www.postgresql.org/about..>

20. **Tobalina, Rubén González Blanco y Sergio Pérez.** *Introducción a Rational Rose*. Universidad de Catalunya : s.n.
21. **Hensgen, Paul.** Manual de Umbrello UML Modeller. [En línea] [Citado el: 16 de enero de 2011.] <http://docs.kde.org/stable/es/kdesdk/umbrello.html>.
22. **Álvarez, Miguel Ángel.** . Lenguaje HTML. [En línea] <http://www.desarrolloweb.com/articulos/711.php>.
23. Programación en Javascript. [En línea] 2009. [Citado el: 1 de febrero de 2011.] [http://www.programacion.com/articulo/curso\\_de\\_javascript\\_1\\_2\\_33](http://www.programacion.com/articulo/curso_de_javascript_1_2_33).
24. Programación en Python. . [En línea] 2009. [Citado el: 27 de enero de 2011.] [http://www.programacion.com/articulo/curso\\_de\\_python..](http://www.programacion.com/articulo/curso_de_python..)
25. Programación en PHP. [En línea] 2009. [Citado el: 29 de enero de 2011.] <http://www.desarrolloweb.com/manuales/12>.
26. Introducción al Lenguaje ASP. [En línea] [Citado el: 29 de enero de 2011.] <http://www.webnova.com.ar/articulo.php?recurso=95>.
27. Sitio de symfony. [En línea] <http://www.symfony.es/>.
28. Ext JS y frameworks Javascript. [En línea] 2009. <http://www.slideshare.net/almarag/ext-js-y-frameworks-javascript..>
29. Sitio oficial de PostgreSQL. [En línea] 2010. <http://www.postgresql.org/about>.
30. Página principal de NetBeans. [En línea] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
31. **Studio., Zend.** Página principal de Zend Studio. [En línea] 2007. [Citado el: 3 de febrero de 2011.] <http://www.tufuncion.com/zend-studio>.
32. **Jose M. Vazquez .** [En línea] 2009. [http://jose.vazquez.be/aptana\\_studio\\_ide\\_javascript\\_php/](http://jose.vazquez.be/aptana_studio_ide_javascript_php/).
33. **Dominio.** 2008.
34. **Quesada., Juan Antonio López.** WRITING EFFECTIV E USES CASE.
35. **Burbeck, Steve.** “Application programming in Smalltalk-80: How to use Model-View-Controller (MVC)”. . [En línea] University of Illinois in Urbana-Champaign. <http://st-www.cs.uiuc.edu/users/smarch/st-doc>.
36. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* **Reynoso, Carlos y Kicillof, Nicolás.** UNIVERSIDAD DE BUENOS AIRES : s.n.
37. **UCI.** Material de caja blanca y caja negra. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 28 de abril de 2011.] [http://eva.uci.cu/file.php/259/Curso\\_2010-2011/Semana\\_9/Conferencia\\_7/Materiales\\_Complementarios/Material\\_de\\_caja\\_b\\_y\\_caja\\_n.pdf..](http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Complementarios/Material_de_caja_b_y_caja_n.pdf..)

## Glosario de Términos

**Blogs:** Es un sitio web que se actualiza periódicamente que recopila cronológicamente artículo de uno o varios autores.

**CMS:** Sistema de soporte de gestión de contenidos.

**Framework:** Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

**TIC:** Son un conjunto de servicios, redes, software que se integran a un sistema de información interconectado y complementario.

**Java:** Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

**Javascript:** Es un lenguaje de programación interpretado, es decir no requiere de compilación.

**PHP:** Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor.

**ASP:** Tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente.

**AJAX:** Técnica de desarrollo web para crear aplicaciones interactivas. Javascript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax.

**UML:** Lenguaje de modelado para especificar o describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

**OMMMA-L:** Extensión del lenguaje de modelado UML Orientada a Objetos de Aplicaciones Multimedia.

**HTML:** Lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**XML:** Es un metalenguaje extensible de etiquetas.

**JQuery:** Framework de Javascript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

**Prototype:** Biblioteca Javascript que permite el uso de controles AJAX y otros efectos visuales en una página web.

Herramienta de Administración de Cuentos del producto “Mis Mejores Cuentos”.

**Scriptaculous:** Biblioteca Javascript que permite el uso de controles AJAX, y otros efectos visuales en una página web.

**IDE:** Entorno de desarrollo integrado.

**CSS:** Lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, siendo imprescindible para crear páginas web complejas.

**Symfony:** Es un framework diseñado para optimizar el desarrollo de aplicaciones web.

**FTP:** Es un sistema de protocolos de transferencias de archivos en un entorno de red.

**JSP:** Es una tecnología Java que permite generar contenido dinámico para la web en forma de documentos HTML, XML u otros.

**MVC:** Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.